

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

PŘÍPRAVEK PRO ŘÍZENÍ PERISTALTICKÉ PUMPY

PERISTALTIC PUMP CONTROLL UNIT

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Jakub Charvát

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Adam Ligocki

BRNO 2021

Bakalářská práce

bakalářský studijní program **Automatizační a měřicí technika**

Ústav automatizace a měřicí techniky

Student: Jakub Charvát

ID: 211424

Ročník: 3

Akademický rok: 2020/21

NÁZEV TÉMATU:

Přípravek pro řízení peristaltické pumpy

POKYNY PRO VYPRACOVÁNÍ:

- 1) Proveďte rešerši dostupných peristaltických pump na trhu. Po konzultaci s vedoucím vhodně zvolte zařízení s požadovanými parametry.
- 2) Navrhněte kompletní architekturu realizovaného systému. Popište použité snímače, komunikační sběrnice a protokoly pro výměnu dat.
- 3) Zvolte vhodnou výpočetní jednotku pro realizaci funkcionality řízení pumpy a komunikace se snímači a centrálním počítačem. Komunikaci mikrokontroléru, snímačů a PC otestujte na demonstrační aplikaci.
- 4) Vytvořte fyzickou realizace zařízení včetně osazení mikrokontroléru, všech dříve vybraných snímačů, napájení a otestujte základní funkcionality všech prvků.
- 5) Vytvořte a otestujte program pro řízení peristaltické pumpy, včetně funkcionalit přesného dávkování kapaliny v čase a všech navržených ochranných mechanismů.
- 6) Vytvořte rozhraní na přídruženém PC a otestujte funkci ovládání a výměny a vizualizace dat mezi PC a Vámi vytvořeným zařízením.

DOPORUČENÁ LITERATURA:

Everett, H.R., 1995. Sensors for mobile robots. CRC Press

Termín zadání: 8.2.2021

Termín odevzdání: 24.5.2021

Vedoucí práce: Ing. Adam Ligocki

doc. Ing. Václav Jirsík, CSc.
předseda rady studijního programu

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Práce se zabývá problematikou peristaltických pump a jsou v ní shromážděny teoretické podklady týkající se tohoto tématu. Primárním cílem práce je realizace vlastní peristaltické pumpy a implementace ochrany před poškozením zařízení přetlakem. Systém vyhodnocování přetlaku zde funguje na principu měření proudu motoru pumpy. V práci je tato premisa úspěšně otestována a aplikována do zařízení. Dalším z cílů práce je vytvoření počítačové aplikace, která zajistí monitorování a řízení pumpy. Přenos dat zastal komunikační protokol MQTT. Komunikace mezi počítačem a pumpou byla úspěšně otestována.

KLÍČOVÁ SLOVA

Peristaltická pumpa, IoT, C#, ESP-32, MQTT, Arduino, Měření proudu, Bezpečnostní prvky

ABSTRACT

The work deals with the problematic of peristaltic pumps and it collects theoretical background information of this topic. The primary goal of the work is the implementation of peristaltic pump and the implementation of protection against equipment damage by overpressure. The overpressure evaluation system in this thesiss on the principle of current measurement of the motor pump. In this work, this premise is successfully tested and applied to the device. Another goal of this work is to create a desktop application that will monitor and control the pump. he data between the application and the device were successfully transfered by the MQTT communication protocol. The communication between the computer and the pump has been tested successfully.

KEYWORDS

Peristaltic pump, IoT, C, ESP-32, MQTT, Arduino, Current measurement, Safety features

CHARVÁT, Jakub. *Přípravek pro řízení peristaltické pumpy*. Brno, 2021, 60 s. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce: Ing. Adam Ligocky,

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Přípravek pro řízení peristaltické pumpy“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno

.....
podpis autora

PODĚKOVÁNÍ

Rád bych poděkoval především panu Ing. Adamovi Ligockimu za mé vedení při sepisování této bakalářské práce. Jeho podněty a připomínky vedli ke zdárnému dokončení práce. Rád bych také poděkoval kolegům Jiřímu Kubincovi a Bc. Matoušovi Hýblovi za pomoc s 3D tiskem a v neposlední řadě slečnám Markétě Vejdové a Lydii Molíkové za pomoc s gramatickou a slohovou korekcí.

Obsah

Úvod	10
1 Úvod do peristaltických čerpadel	11
1.1 Konstrukce	11
1.2 Princip	11
1.3 Klíčové parametry	12
1.3.1 Chemická odolnost	12
1.3.2 Okluze	12
1.3.3 Průtok	12
1.3.4 Pulzace	13
1.4 Použití	14
1.5 Druhy	14
1.5.1 Hadicová čerpadla	14
1.5.2 Trubková čerpadla	15
1.5.3 Mikrofluidní čerpadla	15
2 Komunikační sběrnice/protokoly	16
2.1 MQTT komunikace	16
2.1.1 MQTT klient	16
2.1.2 MQTT server	16
2.2 I2C komunikace	17
2.2.1 Formát zprávy	17
2.3 UART komunikace	18
2.3.1 Formát zprávy	18
3 Příprava podkladů pro realizaci zařízení	19
3.1 Měření proudu pumpy	19
3.1.1 Průběh měření	19
3.1.2 Naměřené průběhy	20
3.2 Architektura zařízení	22
3.2.1 Mikroprocesor	22
3.2.2 Výkonové řízení	23
3.2.3 Proudový snímač	23
3.2.4 Hlava Peristaltické pumpy	23
3.2.5 Display	23
3.2.6 Ovládací prvky	23
3.2.7 MQTT klient	23
3.3 Výběr komponent	24
3.3.1 Výběr peristaltické hlavy	24
3.3.2 Výběr Procesoru	24
3.3.3 Výběr snímače elektrických veličin	25
3.3.4 Výběr interaktivních prvků	26
3.3.5 Výběr výkonového řízení	26
3.4 Ověření komunikace mezi PC a mikroprocesorem	27

3.4.1	Tvorba MQTT klienta na mikroprocesoru	27
3.4.2	Tvorba MQTT klienta na PC	28
3.4.3	MQTT Server	28
4	Program zařízení	29
4.1	Představení platformy Arduino	29
4.2	Představení fenoménu Internet věcí	29
4.3	Konstrukce programu	30
4.4	Virtuální paralelizace - subrutiny	31
4.5	Třídy programu	33
4.5.1	Třída Pump	33
4.5.2	Třída Controls	33
4.5.3	Třída MQTT	33
4.5.4	Třída Display	34
4.5.5	Třída Alarm	34
4.6	Algoritmy řízení	35
4.6.1	Plně manuální mód	35
4.6.2	Polo-manuální mód	35
4.6.3	Automatický mód	36
4.6.4	Intervalový mód	36
5	Ovládací aplikace	37
5.1	Účel a funkce aplikace	37
5.2	Problematika návrhu uživatelského rozhraní	38
5.3	Představení frameworku Windows Presentation Foundation	38
5.4	Architektura aplikace	39
5.4.1	MainWindow	40
5.4.2	MonitorPage	41
5.4.3	SettingsPage	42
5.4.4	MQTTPage	43
5.4.5	Pohyb v aplikaci	44
6	Realizace zařízení	45
6.1	Deska plošných spojů	45
6.2	Grafické rozhraní zařízení	47
6.3	Návrh a výroba šasi	48
7	Zhodnocení dosažených výsledků	49
7.1	Zařízení peristaltické pumpy	49
7.2	Ovládací aplikace	49
7.3	Dosažené výsledky vzhledem k základnímu zadání	50
	Závěr	51
	Literatura	52
	Seznam symbolů, veličin a zkratk	54

Seznam příloh	55
A Příloha - Komunikace pumpy a aplikace	56
B Příloha - Fotografie zařízení	57
C Obsah přiloženého CD	60

Seznam obrázků

1.1	Díly peristaltické pumpy	11
1.2	Princip funkce	11
1.3	Rychlost proudění bez tlumiče	13
1.4	Rychlost proudění s tlumičem	13
1.5	Hadicové peristaltické čerpadlo	14
1.6	Trubková peristaltická čerpadla	15
1.7	Mikrofluidní čerpadlo	15
2.1	MQTT protokol	16
2.2	I2C komunikace	17
2.3	Formát zprávy I2C protokolu	17
2.4	UART Komunikace	18
2.5	Formát UART zprávy	18
3.1	Schéma měření	19
3.2	Průběh bez zátěže	20
3.3	Průběh při podtlaku	20
3.4	Průběh při přetlaku	21
3.5	Blokové schéma	22
3.6	Základní schéma zapojení snímače INA-219	25
3.7	Aplikace MQTT klienta	28
4.1	Konstrukce programu	30
4.2	Posloupnost subrutin	31
4.3	Vývojový diagram zpožďujících sekvencí	32
4.4	Seznam prvků menu pumpy	34
4.5	Vývojové diagramy základních módů pumpy	35
4.6	Vývojové diagramy pokročilých řídicích módů pumpy	36
5.1	Architektura aplikace	39
5.2	MainWindow	40
5.3	MonitorPage	41
5.4	SettingsPage	42
5.5	MQTTPage	43
5.6	Pohyb v aplikaci	44
6.1	Schéma zařízení	45
6.2	Proces návrhu desky plošných spojů.	46
6.3	GUI zařízení	47
6.4	Výsledné rendery zařízení	48
7.1	Porovnání vzhledu pump	49
A.1	Propojení aplikace a pumpy	56
B.1	Alenčino zrcadlo	57
B.2	Alenčino zrcadlo	57
B.3	Alenčino zrcadlo	58
B.4	Alenčino zrcadlo	58
B.5	Alenčino zrcadlo	59
B.6	Alenčino zrcadlo	59

Úvod

Peristaltické pumpy jsou pumpy, které fungují na principu přenášení kapalin podtlakem. Podtlak v pumpě vytváří rotující válečky, které posouvají kapalinu. Protože tyto pumpy dokáží přenášet kapalinu asepticky, čistě a přesně, mají nenahraditelné zastoupení v oblastech jako je farmaceutický, potravinářský nebo chemický průmysl.

I přes nesporné výhody mají pumpy tohoto typu nedostatky, a to především ve stlačování pružné trubice, ve které je podtlak tvořen. Mimo to jsou také velice náchylné k poškození přetlakem. Pružné trubice jsou z velice jemných materiálů, a proto jsou v případě přetlaku z hlediska odolnosti nejslabším článkem ústrojí.

Tato bakalářská práce se bude zabývat především návrhem a výrobou přípravku pro řízení peristaltické pumpy, který odstraní právě tento nedostatek. Premisou pro řešení tohoto problému je monitorování proudu motorem pumpy, v případě že se proud zvýší, můžeme předpokládat, že se motor více namáhá právě kvůli přetlaku, důsledkem čehož by mohlo dojít k poškození zařízení a v horším případě i poranění obsluhy, pokud by byla pumpována agresivní kyselina.

Po prozkoumání dostupných informací od výrobců peristaltických čerpadel bylo zjištěno, že většina pump takovouto ochranu nemá a pokud ano, tak je implementována ve velice nákladných modelech.

Další inovací, kterou by mělo navrhované zařízení přinést, je bezdrátová konektivita, která dovolí přidružené aplikaci sledovat stav pumpy, ovládat ji a vzdáleně upravovat její nastavení. Bezdrátová konektivita také přináší předpoklad k propojování jednotlivých pump mezi sebou. Díky tomu by mohly tvořit komplexní dávkovací stanice s různými druhy kapalin.

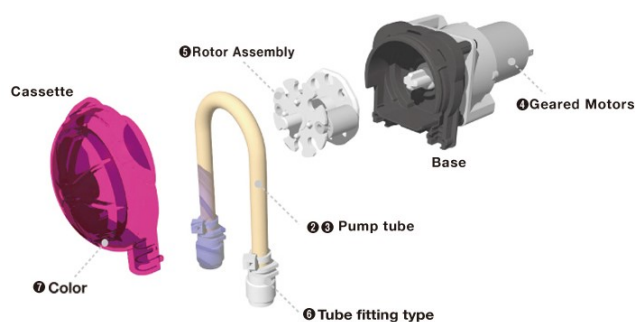
1 Úvod do peristaltických čerpadel

Peristaltická čerpadla jsou velice přesná čerpadla, která mají velké množství výhod pro specifická použití, jež jsou rozebrány v kapitole 1.4. V této kapitole bude rozebrána jejich konstrukce, princip funkčnosti, klíčové parametry, které nejvíce ovlivňují funkci a životnost čerpadla, použití a druhy.

1.1 Konstrukce

Peristaltická pumpa je zpravidla zkonstruována z hlavice pumpy a elektromotoru. Hlavu tvoří základna, ve které je vedena pružná trubice s fitinkami vedoucí kapalinu, dva nebo více přitlačných válečků a přední kryt, to je ilustrativně zobrazeno na obrázku 1.1.

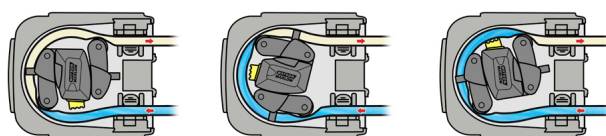
Typ elektromotoru je variabilní v závislosti na požadovaném použití. Nejběžnějším typem je klasický DC motor, pokud je požadována vysoká přesnost, je výhodné zvolit krokový motor a pokud je nutné dávkovat média s vyšší hustotou než je voda, můžeme využít střídavého bezkartáčového motoru s převodem. Alternativně je možné též použít jakýkoli zdroj rotačního pohybu, např. vzduchem poháněné motory pro výbušné prostředí, nebo benzinový motor pro pumpy o velkém výkonu.



Obr. 1.1: Peristaltická pumpa rozdělená na její elementární prvky. [1]

1.2 Princip

Chod peristaltické pumpy funguje na principu podtlaku, kdy přitlačný váleček v prvním kroku přitlačí pružnou trubičku ke stěně základny. V druhém kroku tento přitlačený bod táhne po stěně základny. Tím vzniká v uvolňované části trubičky podtlak, který je vyrovnán nasátím kapaliny. První přitlačný váleček vede kapalinu k výstupnímu otvoru. Ve třetím kroku uvolní přitlačný váleček cestu kapalině a další přitlačný váleček začíná tlačit na pružnou přívodní trubičku. Následně se cyklus opakuje. Demonstrace dávkovacího cyklu je znázorněna na obrázku 1.2.



Obr. 1.2: Ilustrativní postup znázornění proudění média trubičkou pumpy. [2]

1.3 Klíčové parametry

Tato kapitola bude pojednávat o důležitých parametrech, na které je nutné při vývoji zařízení brát ohled. Budou zmíněny v kapitole 3.3, na jejichž základě bude vybrána pumpa použitá při realizaci zařízení.

1.3.1 Chemická odolnost

Požadovaná chemická odolnost se zde vztahuje jen na pružnou trubičku, skrz kterou je vedeno médium. Tato trubička musí splňovat požadavek na zachování pravidelného kruhového průměru po miliony cyklů stlačení a roztahení. Požadavek na elasticnost materiálu vylučuje veškeré neelastomerové polymery, jako je například PTFE. Ideálními materiály pro pružnou trubičku může být například silikon, PVC nebo EPDM guma. V dnešní době se experimentuje s kombinací elastomerových a neelastomerových polymerů, kde se neelastomer použije jako vložka uvnitř trubičky a trubička samotná je tvořena elastomerem s vyšší pružností a nižší chemickou odolností. Hlavním důvodem je zvýšení životnosti trubiček. [3]

1.3.2 Okluze

Okluze je termín, který se využívá pro měření množstevního stlačení. Zde konkrétně vyjadřuje minimální mezeru mezi přitlačným válečkem a stěnou základny. Pokud je okluze příliš velká, je dramaticky snižována životnost trubiček. Pokud je okluze příliš malá, může dojít k nedovření trubice a vytékání média zpět do přívodní trubičky vlivem malého podtlaku v pumpě. Právě proto je okluze velice zásadním konstrukčním parametrem pump a často bývá uváděna v procentech, typicky 10 až 20 %. Vzorec uvedený níže vyjadřuje okluzi jako podíl vnitřní mezery v trubičce při stlačení a dvojnásobku tloušťky stěny hadičky. [3]

$$y = \frac{2t - g}{2t} * 100 \quad [\%]$$

y - okluze, g - mezera mezi válečkem a základnou, t - tloušťka stěny trubičky

1.3.3 Průtok

Průtok je jedním z nejdůležitějších parametrů pumpy pro spotřebitele.

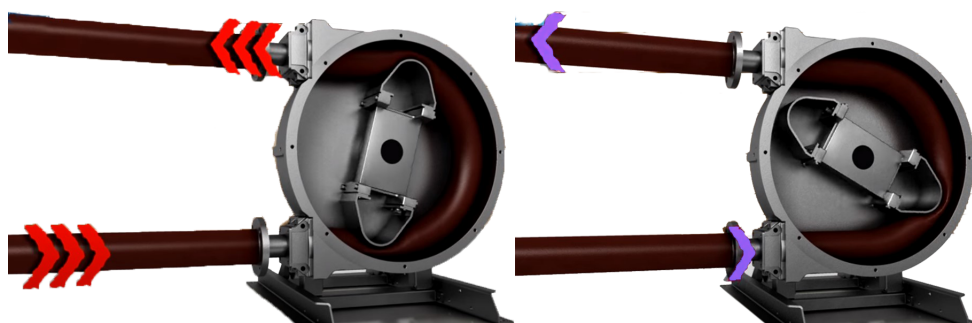
Je ovlivněn mnoha výrobními parametry, jako například:

- Vnitřní průměr trubky
- Vnější průměr hlavy pumpy
- Otáčky hlavy pumpy
- Vstupní pulsace

Zvýšení počtu válečků nezvýší průtok, namísto toho sníží efektivitu, neboť ovlivní aktivní délku hadičky. Zvýšení počtu válečků taktéž zvýší pulsaci na výstupu pumpy. Průtok peristaltického čerpadla není lineární a prakticky vždy se na výstupu objevují pulsy. Tyto pulsy ovlivňují přesné měření dávkování média, a proto může být přesného měření dosaženo jen s lineárním průtokem. Eliminování pulsace lze dosáhnout se správně navrženými pulsními tlumiči (rozebráno v kapitole 1.3.4 Pulzace) [3].

1.3.4 Pulzace

Pulsace je vedlejším negativním účinkem peristaltického čerpadla. Neboť při nabrání kapaliny do hadičky válečkem vznikne uzavřený objem kapaliny, který je pod tlakem, a při přivedení k výstupu hlavice je uvolněn a vytvoří tak puls kapaliny, který je nežádoucí. Je-li médium takto vymršťováno do pevné ukotvené trubky, mohou zde vznikat vibrace, které opotřebovávají uchycení trubek. Pokud je výstupní trubička připojena k pružné trubičce, je na jejím výstupu znatelný třes. Nehledě na tom, že výstup v jakémkoliv vedení bude nelineární. Tyto vibrace lze odstranit takzvanými tlumiči pulzů, ty jsou realizovány zásobníkem, který je T spojkou připojen jak k vývodu pumpy, tak ke vstupu do připojeného potrubí. Jeho přítomnost vyrovná tlak v trubičce tím, že přebytečné médium v sobě uschová a následně z něj odteče poté, co tlak klesne. Funkci můžeme připodobnit k vyhlazovacímu kondenzátoru v elektrickém obvodu. [3]



Obr. 1.3: Šipky na obrázku reprezentují rychlost proudění média potrubím v případě absence tlumiče. Červené šipky (rychlé proudění) jsou ve stavu, kdy je médium nasáváno a vytlačováno z hlavy pumpy. Fialové šipky (pomalé proudění) jsou ve stavu kdy nedochází k maximu rychlosti kapaliny pumpou. Tyto změny rychlostí mají za výsledek pulzaci média na výstupu.[4]



Obr. 1.4: Šipky na obrázku znázorňují rychlost proudění média při použití tlumiče na vstupu i na výstupu, tlumič zajistí konstantní rychlost proudění média potrubím a snižuje pulzaci.[4]

1.4 Použití

Použití peristaltických pump je velice výhodné kdekoli, kde potřebujeme přesně odměřené množství média, které je častokrát velice agresivní zásadou či kyselinou. Neopomenutelnou výhodou použití peristaltických pump je předcházení kontaminace média, neboť médium nepřichází do přímého kontaktu s materiálem pumpy a je stále uzavřeno v trubičkách. Tato specifikace dává pumpám ideální předpoklad pro použití v různých odvětvích, jako je například farmacie, potravinářství či obecně chemický průmysl. [3]

Z typických příkladů můžeme konkrétně jmenovat:

- **Zdravotnictví**
 - Dialyzační přístroje
 - Lékařské infuzní pumpy
 - Farmaceutická výroba
- **Zemědělství**
 - Čerpadla pro extrakci javorové mízy
 - Dávkovače pro hydroponické systémy
- **Strojírenství a výroba**
 - Čerpadlo na beton
 - Celulóza a papírenské závody
 - Inkoustové tiskárny

1.5 Druhy

Vzhledem k tomu, že některé aplikace čerpadel vyžadovaly parametry na hranici realizovatelnosti tradičních čerpadel, byly vyvinuty další typy čerpadel, než jen základní trubkový typ. Další typy jsou rozebrány a vysvětleny v této kapitole.

1.5.1 Hadicová čerpadla

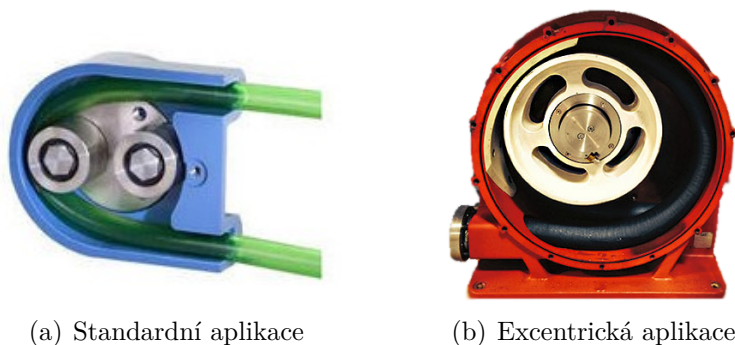
Hadicová čerpadla bývají větší než typické peristaltické pumpy. Hlavním rozdílem oproti konvenčnímu typu je ten, že místo přitlačných rotačních válečků používají pouze přitlačná kluzná ramena, která nerotují kolem své vlastní osy, ale jsou pevně spojena se středovou hřídelí. Tato konstrukce ve spojení s větším průměrem hadic dosahuje tlaku až 16 barů. Hlava pumpy bývá vyplněna mazivem, aby se prodloužila životnost trubek, která je absencí přitlačných válečků razantně zkrácena. Pokud není požadován vysoký tlak, je zpravidla lepší použít trubková čerpadla. [3]



Obr. 1.5: Hadicové peristaltické čerpadlo. [5]

1.5.2 Trubková čerpadla

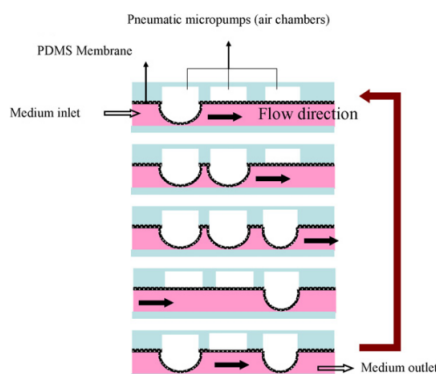
Nízkotlaké pumpy, které k vytvoření okluze využívají přitlačné válečky, které mohou rotovat kolem vlastní osy. Díky tomu nemusí být útroby peristaltických pump mazány mazivem. S výjimkou excentrické jednoválečkové pumpy mívá tento typ minimálně 2 válečky, které periodicky rotují kolem hlavní osy a zajišťují přepravu kapaliny. Válečků může být téměř neomezený počet, nicméně nejvyšší používaný počet je 12. Zvýšení počtu válečků zvyšuje pulzaci na výstupu a snižuje amplitudu pulsu. Nevýhodou příliš velkého počtu válečků je zvýšení počtu stlačení trubky, což úměrně počtu válečků snižuje životnost trubiček. [3]



Obr. 1.6: Trubková peristaltická čerpadla [6] a [7]

1.5.3 Mikrofluidní čerpadla

Konvenční pumpy nedokáží dávkovat malé objemy, což může být problém například v analytické chemii nebo v oblasti mikrofluidiky, kde jsou dávkované objemy v řádech mikrolitrů a nižších. Mikrofluidní čerpadla byla vyvinuta pro minimalizaci cirkulačního objemu čerpaného média. Jedná se o zařízení, jehož struktura je přímo implementována do mikrofluidního obvodu a stává se tak jeho interní součástí, nikoliv externí jako u předešlých typů pump. Zařízení funguje na principu několika miniaturních pneumatických membránových pístů, které po aktivaci vtlačí svou membránu do řečiště dávkovaného média. Při sekvenční aktivaci těchto pístů může pumpa posouvat malé množství média ve směru toku média. [3]



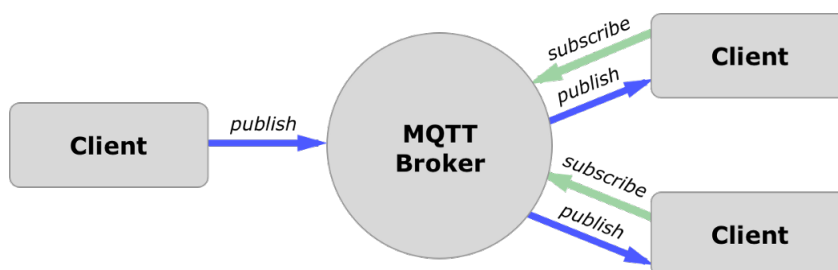
Obr. 1.7: Diagram postupu pístů mikrofluidního čerpadla k posuvu kapaliny. Postupnou deformací membrány vznikají nízkoobjemové komory. Skrze ně je možné transportovat malé množství kapaliny. [8]

2 Komunikační sběrnice/protokoly

Z důvodu různorodosti ve způsobu komunikace různých částí zařízení je nutné použít různé způsoby komunikace s nimi. V této kapitole budou rozebrány všechny použité komunikační sběrnice a protokoly a budou vysvětleny jejich základní principy a formy, jakými komunikují s okolím. Kapitola vychází ze zdroje [18].

2.1 MQTT komunikace

MQTT (Message Queue Telemetry Transport) je jednoduchý komunikační protokol, jenž funguje na principu Publish/Subscribe (čili publikování a odběr zpráv) komunikace mezi MQTT klienty přes MQTT server čili zprostředkovatele. Pokud má odesílatel nová data k odeslání, odešle je serveru v příslušném formátu. Server následně distribuuje tuto zprávu všem klientům, kteří dané téma odebírají. Klient, který tato data odeslal, nemusí mít žádné informace o tom, kterým dalším klientům je předá, veškerou distribuci zajistí server.



Obr. 2.1: Ilustrace principu MQTT komunikace, kde jeden ze tří klientů pouze odesílá informace a dva další klienti informace odesílají i přijímají. [11]

2.1.1 MQTT klient

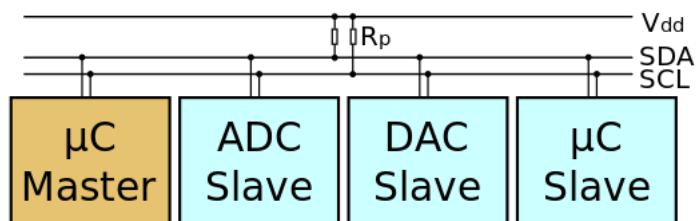
Klientem může být jakékoliv zařízení od mikrokontrolérů, přes desktopové aplikace až po servery. Zařízení spouštějí MQTT knihovny a pomocí sítě se připojují k brokeru. V případě, že se klient k serveru připojí poprvé, nastaví zprávu, kterou dostanou všichni odběratelé v případě, že se klient neočekávaně odpojí od serveru. Zpráva v protokolu MQTT může mít minimálně 2b dat a maximálně až 256 Mb.

2.1.2 MQTT server

Server funguje v zásadě jako pošta, jen pro adresování nepoužívá konkrétní adresu ale, předmět s názvem **téma** (topic) a každý, kdo má zájem o tuto zprávu, se přihlásí k odebírání tohoto tématu. Jeden odesílatel tak může publikovat pro mnoho odběratelů, stejně tak jako mnoho odesílatelů může dodávat zprávy jen jednomu odběrateli. Už díky tomuto principu centralizace zpráv poskytuje server jistou dávku zabezpečení, neboť jednotliví klienti nemají žádné informace o zbylých klientech. Další vrstva zabezpečení je šifrování TLS (Transport Layer Security) s uživatelským jménem, heslem a případně i certifikacemi, které server vyžaduje po klientovi, aby mu umožnil připojení.

2.2 I2C komunikace

Tento komunikační protokol funguje na principu Master/Slave, kdy je minimálně jedno zařízení nastaveno jako Master a libovolný počet zařízení jako Slave (omezen šířkou sběrnice). Je používán především pro komunikaci s periferiemi, které nemají extrémní požadavky na rychlost. I2C sběrnice je velmi náchylná na rušení, protože je v ní přenášeno relativně velké množství informací, a proto se zpravidla používá pro komunikace na krátkou vzdálenost, ideálně pouze na desce plošných spojů. [12]. Na obrázcích 2.2 a 2.3 je vyobrazeno blokové schéma zapojení několika zařízení a formát zprávy, kterým komunikují.



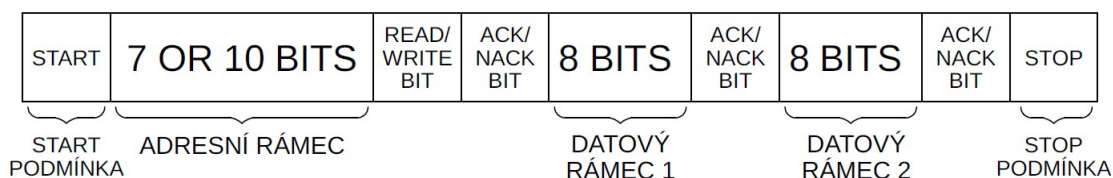
Obr. 2.2: Bloková ilustrace I2C zapojení jednoho zařízení v roli Master a tří Slave zařízení [16].

Princip

Protokol používá jednu obousměrnou datovou linku (SDA-Serial Data Line) a jednu jednosměrnou časovací linku (SCL – Serial Clock Line). Následně jsou zde připojeny i napájecí vodiče +V a GND. Hodnota napájecího napětí odpovídá hodnotě, jež je v použitém systému definována jako logická 1 (typicky +5V nebo +3,3V). Master generuje hodinový signál pro ostatní zařízení a inicializuje komunikaci s nimi, Slave přijímá hodinový signál a reaguje na požadavky od Mastera, pokud je adresován. [12]

2.2.1 Formát zprávy

- **Start podmínka** – SCL v log. 1 a SDA přechází z log. 1 do 0
- **Read/Write bit**
 - log. 0 -> Master posílá data do Slave zařízení
 - log. 1 -> Master přijímá data od Slave zařízení
- **ACK/NACK bit** – nese informaci o korektně přijatém datovém nebo adresním rámci, pokud příjemce korektně přečetl data, přidrží SDA v log. 0.
- **Stop podmínka** – SCL v log. 1 a SDA přechází z log. 0 do 1



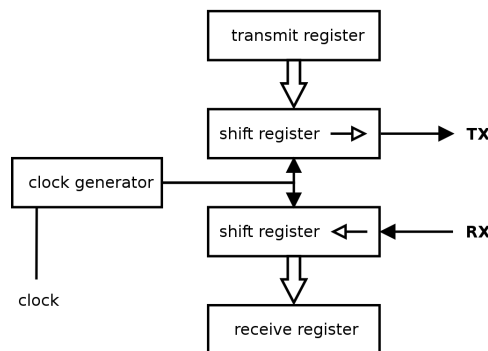
Obr. 2.3: Formát zprávy I2C protokolu, obrázek vychází ze zdroje [17].

2.3 UART komunikace

Komunikační protokol UART (Universal Asynchronous receiver-transmitter) je hardwarové zařízení pro asynchronní sériovou komunikaci. Formát komunikace i její rychlost je zde nastavitelná podle požadavků uživatele. Protože je UART hardwarové zařízení, musí být fyzicky implementováno do obvodů, které mají touto komunikací disponovat. [12]

Princip

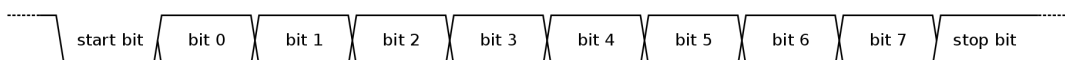
UART odesílá jednotlivé bity sekvenčním způsobem za pomoci posuvného registru a v cíli jej druhý UART přijímá také do posuvného registru. Díky těmto registrům není nutné synchronizovat hodinový signál mezi dvěma zařízeními, ale jen rychlost přijímání dat (aby na jedné nebo druhé straně nedocházelo k přehlcení registru). Rychlost komunikace sběrnice UART se nastavuje v tzv. baudech (1bit/s). Typické hodnoty rychlosti komunikace jsou 1200, 2400, 4800, 9600 115200. Běžně se používá 9600. Ačkoliv je v porovnání s paralelním odesíláním dat sériové řazení bitů pomalejší, je značně levnější kvůli redukci počtu kabelů požadovaných na paralelní zapojení. UART umožňuje komunikovat v jednom směru (simplex), v obou směrech najednou (duplex) nebo střídání vysílání a příjmu (poloviční duplex).



Obr. 2.4: Blokové schéma sběrnice UART se znázorněním funkce posuvných registrů v této sběrnici [17].

2.3.1 Formát zprávy

Každý datový rámec začíná start bitem a končí stop bitem. Mezi těmito signály je pak datový rámec. Po přenesení datového rámce je možnost kontrolovat bit parity, která je však volitelná. Parita může být lichá nebo sudá, což je určeno před započítím konverzace. Pokud je parita sudá, paritní bit obsahuje log. 1, v případě že je v datovém rámci sudý počet log. 1. Pokud je parita lichá, paritní bit obsahuje log. 1, v případě, že je v datovém rámci lichý počet log. 1.



Obr. 2.5: Ukázka formátu zprávy UART sběrnice. [13]

3 Příprava podkladů pro realizaci zařízení

Tato kapitola, sdružující podklady nutné pro realizaci zařízení, se zabývá především výběrem komponent, ze kterých se bude zařízení skládat, jeho architekturou s blokovým schématem a v neposlední řadě měřením proudu do motoru v několika scénářích, ve kterých se zařízení může nacházet.

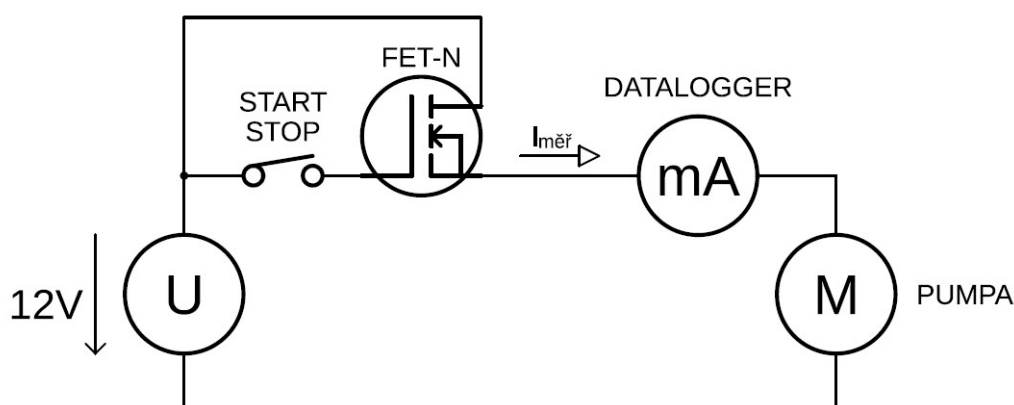
3.1 Měření proudu pumpy

Dříve než byly vybrány klíčové komponenty, bylo provedeno jednoduché měření proudu do pumpy při různých scénářích jejich kritických stavů, které jsou hlavní motivem této práce.

3.1.1 Průběh měření

Jako zdroj napájení byl použit standardní laboratorní zdroj se stabilizovaným výstupem nastaveným na pracovní napětí motoru, jež bylo 12V. Aby mohla být pumpa zapnuta a vypnuta bez nutnosti vypínání zdroje, byl do obvodu přidán modul s MOS FET tranzistorem typu N, který slouží jako výkonový spínací prvek. Schéma měřicího obvodu, kterým byly průběhy měřeny, je na obrázku. 4.1.

Funkci měřicího zařízení zde zastoupil multimetr OWON OW18B nastavený do režimu miliampérmetru, který dokáže měřená data odesílat přes bluetooth k dalšímu zpracování. Pomocí mobilní aplikace od výrobce multimetru bylo možné zaznamenat průběhy do databáze k dalšímu zpracování. Zpracování měření bude rozebráno v následující kapitole 4.1.2 Naměřené průběhy.

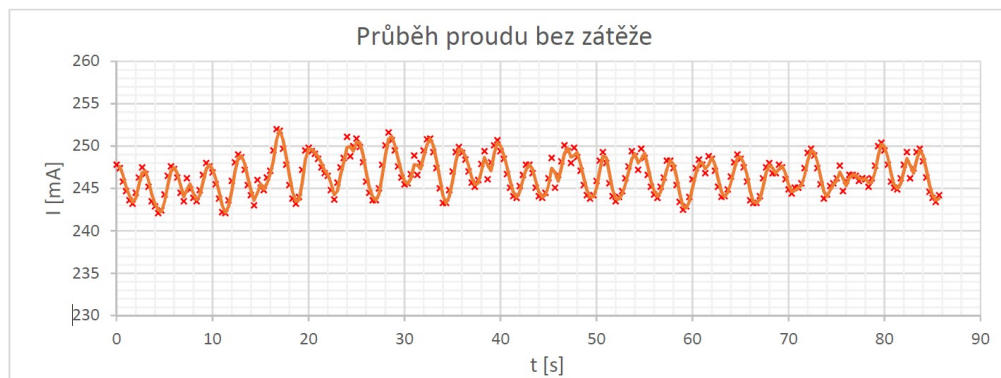


Obr. 3.1: Schéma zapojení zdroje, pumpy a miliampérmetru, pro měření průběhu proudu do motoru pumpy.

3.1.2 Naměřené průběhy

U měření byl přenášným médiem vzduch. Měření proběhlo na pumpě firmy Welco modelu WPM.

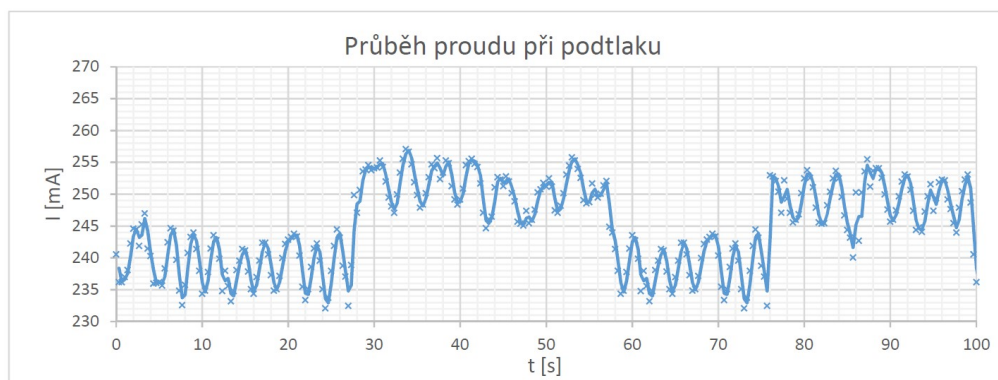
Bez zátěže



Obr. 3.2: Záznam průběhu proudu do motoru v případě standardního provozu.

První měření zaznamenávalo průběh proudu do pumpy bez omezení přívodní i odvodní trubičky. Minimální hodnota byla 242mA, maximální hodnota byla 252mA a přibližná průměrná hodnota byla 247mA. Na obrázku 4.2 lze vidět typický zvlněný průběh proudu do motoru pumpy.

Při podtlaku

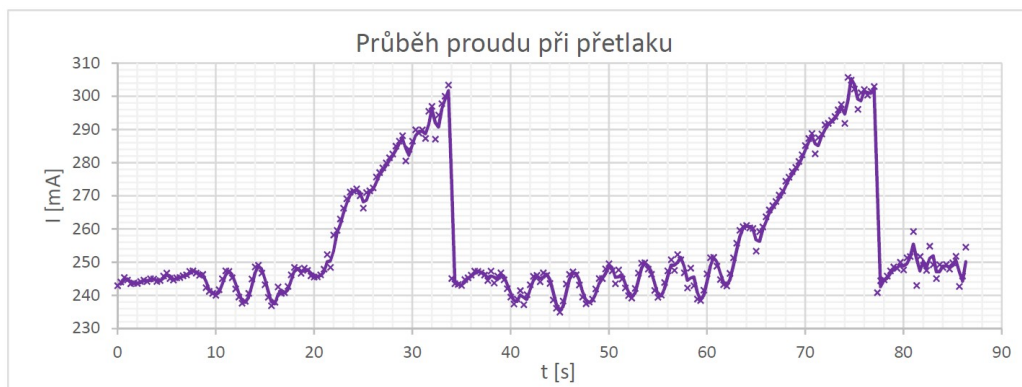


Obr. 3.3: Záznam průběhu proudu do motoru v případě ucpávání přívodní hadičky. V intervalech s nižší střední hodnotou proudu byla trubička ucpána a v důsledku toho se motor pro absenci neseného média nemusel tolik namáhat.

Druhé měření probíhalo tak, že v intervalech přibližně 20 vteřin byla ucpávána a opět uvolňována přívodní trubička pumpy. Pumpa tak neměla nosné médium, čímž vznikl podtlak a motor se nemusel namáhat tolik, aby podtlakem již předem stlačenou trubičku opět stlačil.

Průměrná hodnota průběhu při podtlaku byla 237mA a lze ji na obrázku 4.3 pozorovat v úsecích 0-27 vteřin a 56-76 vteřin. Průměrná hodnota průběhu bez podtlaku byla přibližně 249mA, tento průběh je na grafu zaznamenán na časové ose v úsecích 28-56 vteřin a 77-99 vteřin.

Při přetlaku



Obr. 3.4: Záznam průběhu proudu do motoru v případě ucpávání odvodní hadičky. Na grafu jsou díky strmému nárůstu proudu jasně rozeznatelné oblasti, kde byla trubička ucpána. Změna proudu je oproti obrázku 3.3 několikanásobná a můžeme ji bezpečně detekovat.

Třetí a poslední měření zaznamenávalo průběh v situaci, kdy se ucpe odvodní trubička. Především kvůli tomuto měření musel být použit jako médium vzduch, protože při ucpání odvodní trubičky při pumpování vody téměř okamžitě trubička vyletěla vlivem násobně většího tlaku, než je schopen vyvinout stlačený vzduch. Odvodní trubička byla pouze navlečena na vývodu z pumpy, nikoliv zajištěná. Při pumpování vody by byl průběh ovlivněn pouze strmějším nárůstem proudu.

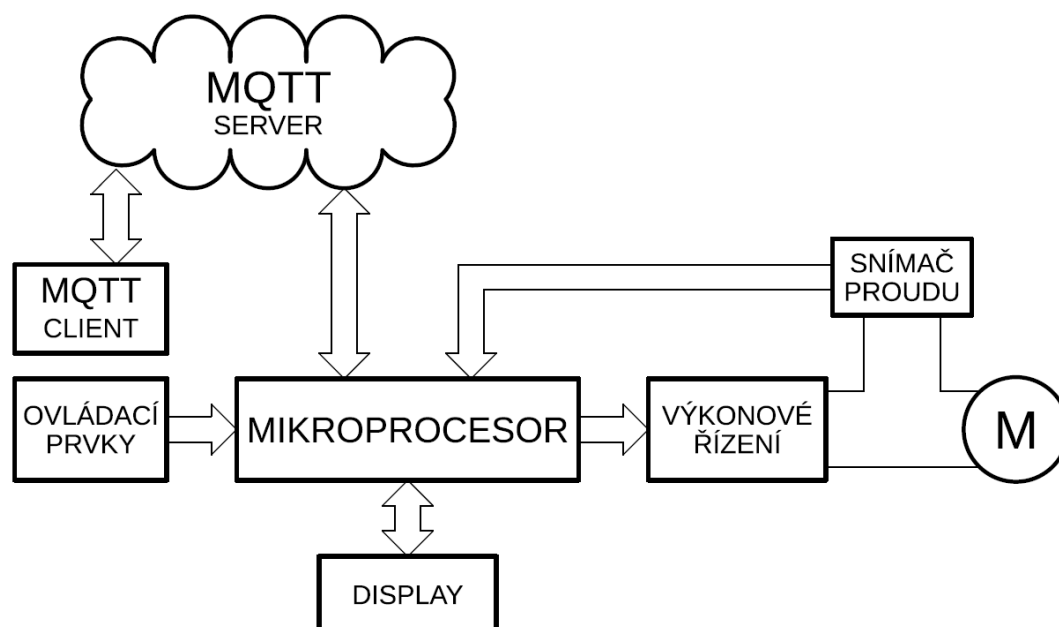
I přes pumpování pouhého vzduchu je oproti předchozím průběhům patrný velice strmý náběh proudu v časových úsecích 20-35 vteřin a 62-78 vteřin na obrázku 4.4. Maximální hodnota, které v měření proud dosáhl, byla 305mA. Po překonání této pomyslné hranice se trubička vlivem tlaku uvolnila. Zbytek průběhu odpovídá standardnímu průběhu bez zátěže s průměrnou hodnotou 245mA.

Závěr z měření

Měření poskytlo průběhy ve třech scénářích: neomezený provoz, provoz při podtlaku a provoz při přetlaku. Potvrdilo základní premisu o stoupání proudu spotřebovaného motorem, na které byla postavena celá myšlenka zařízení. A posteriori bylo zjištěno, jakým způsobem ovlivní průběh proudu podtlak v přívodní trubičce, a že rozdíl je znatelný a měřitelný. Další podstatný výstup z měření udal jednotky, ve kterých se měřená hodnota pohybuje. Měření probíhalo na menším modelu WPM, protože v době měření nebyl model, jenž by měl být v zařízení, dostupný. Základní princip měření byl otestován bez přenášení kapaliny, nicméně jednotky, ve kterých se měření pohybovalo, jsou pro následující zpracování stále relevantní.

3.2 Architektura zařízení

Kapitola popisující jednotlivé bloky blokového schématu (obrázek 3.5). Je zde roze-psána funkce každého podstatného prvku v zařízení, jeho role a způsob, jakým bude komunikovat z okolím.



Obr. 3.5: Blokové schéma architektury zařízení. Šipky znázorňují tok informací mezi bloky. Středem veškerých informačních toků je blok s mikroprocesorem, ten komunikuje s externí aplikací skrz MQTT protokol, interaktivním displejem přes UART, ovládacími prvky přímo připojenými na GPIO vstupy zařízení, snímačem proudu skrz I2C a výkonovým řízením PWM signálem.

3.2.1 Mikroprocesor

Mikroprocesor bude srdcem celého zařízení, bude zajišťovat čtení ze senzorů, ovlivňovat chod peristaltické pumpy, posílat informace na MQTT server a interagovat s uživatelem skrz displej a ovládací prvky. Procesor bude interagovat s ostatními bloky následujícím způsobem:

- Proudový snímač – I2C
- Výkonové řízení motoru – PWM
- Displej – UART
- MQTT server – Wi-Fi
- Ovládací prvky - GPIO

3.2.2 Výkonové řízení

Výkonové řízení bude obsahovat řídicí H-můstek. H-můstek je zařízení, jež je schopné reverzovat směr otáčení motoru a regulovat PWM signálem výkon motoru pumpy. Zařízení se standardně skládá z dvou až čtyř tranzistorů, budičů tranzistorů a součástek nastavujících pracovní bod tranzistorů. H-můstek však není podstatou této práce, a proto nebude podrobněji rozebrán.

3.2.3 Proudový snímač

Snímač bude s nadřazeným systémem komunikovat pomocí I2C komunikační sběrnice. Tento snímač bude nadřazenému systému podávat informace o proudu, napětí i celkovém výkonu na motoru. Nadřazený systém bude díky tomuto snímači oddělen od výkonové měřené veličiny a bude dostávat informace pouze přes I2C, nedojde tak k ohrožení systému přepětím nebo nadproudem.

3.2.4 Hlava Peristaltické pumpy

Peristaltická hlava bude sloužit k dopravě média pouze jako hardwarový prostředek. Motor peristaltické hlavy bude ovládán výstupem z výkonového řízení PWM signálem.

3.2.5 Display

Dotykový displej bude sloužit jako HMI (Human Machine Interface) panel. Bude uživateli zprostředkovávat informace o důležitých měřených parametrech. Zároveň bude zajišťovat funkci ovládacího panelu spolu s mechanickými ovládacími prvky. Displej komunikuje s nadřazeným systémem pomocí UART rozhraní.

3.2.6 Ovládací prvky

Modul s ovládacími prvky bude zajišťovat interakci uživatele a systému standardním mechanickým stylem tlačítek a otočných knoflíků. Modul by měl obsahovat minimálně jeden rotační enkodér a několik tlačítek s rychlou volbou pro rychlý přístup k často používaným funkcím pumpy.

3.2.7 MQTT klient

Aplikace, jež bude komunikovat s peristaltickou pumpou v roli klienta přes MQTT server, bude zapsaná v jazyce C# frameworkem WPF. Z aplikace bude možné komunikovat s pumpou, přímo ji ovládat, odesílat do ní data s instrukcemi, nastavovat ji a přijímat z ní informace.

3.3 Výběr komponent

Zde je kapitola zabývající se výběrem komponent, které vycházejí z předchozího měření v kapitole 3.1, dostupných komponent a inovativních požadavků na pumpu. Při výběru komponent byl brán zřetel na ideální poměr ceny/výkonu (kvality), neboť stále se jedná o prototyp, na kterém je třeba otestovat teoreticky zamýšlené principy.

3.3.1 Výběr peristaltické hlavy

Požadavky na hlavu pumpy:

- **Typ** – Trubkové čerpadlo
- **Průtok** - 200ml/min
- **Oblast použití** - Laboratorní prostředí, dávkování chemikálií
- **Materiál trubiček** - Silikon
- **Motor** - DC Motor

Primárně z důvodu dostupnosti a variability čerpadel bylo zvoleno čerpadlo od japonské společnosti Welco. Vzhledem k velikosti, maximálnímu průtoku 200ml/min a ceně byl vybrán model střední třídy **WPX1**. Hlavní výhodou pump od firmy Welco je jejich jednoduchá údržba a snadná smontovatelnost. Firma Welco poskytuje plno konfigurovatelných prvků, které pumpa obsahuje.

Mezi konfigurovatelné parametry patří například:

- Materiál a tloušťku trubiček
- Typ motoru
- Typ fitinky
- Počet přitlačných válečků

Díky této variabilnosti konfigurace může sám zákazník objednat typ na míru jeho požadavků bez nutnosti vybírání ze sortimentu předkonfigurovaných pump.

3.3.2 Výběr Procesoru

Procesor, který bude vhodný pro aplikaci v zařízení, musí nabídnout dostatek výkonu a dostatek vstupně výstupních GPIO pinů (General purpose input/output), předběžným předpokladem je minimálně 20 GPIO. Zároveň musí podporovat všechny potřebné komunikační protokoly jako je UART, I2C a vzhledem k IoT koncepci zařízení je potřeba vybrat procesor, který dovoluje použití bezdrátové konektivity, jako je WiFi a BLE (Bluetooth Low Energy).

Nejlepší kompromis mezi výkonem, dostupností, parametry a cenou nabízí platforma ESP32, která poskytuje velké množství standardních i specializovaných Embedded systémů. Konkrétní vybraný mikroprocesor je LOLIN32.

Hlavní parametry, které vedly k výběru tohoto modelu:

- Frekvence jádra 80-240 MHZ
- Přítomnost bezdrátové konektivity (WiFi a BLE)
- 34 GPIO pinů

3.3.3 Výběr snímače elektrických veličin

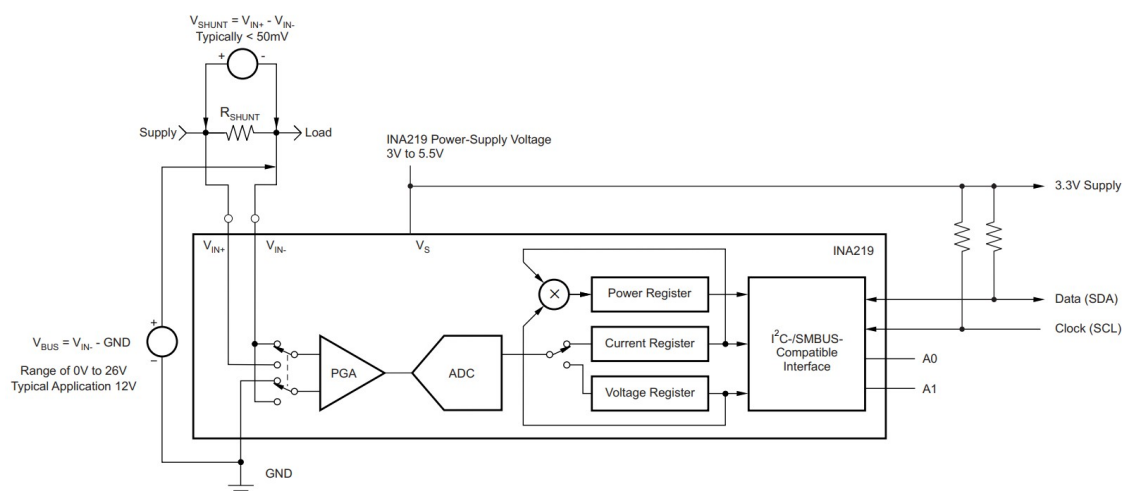
Po rešerši dostupných snímačů proudu byl vybrán snímač **INA-219**. Oproti ostatním snímačům dokáže měřit s velmi vysokou přesností a to až na 1mA. Nutná jemnost měření byla stanovena experimentálním měřením, které je popsáno v kapitole 4.1. Tento rozdíl proudů se pohyboval v řádově desítkách mA.

Snímač proudu dokáže mimo jiné podávat informace i o napětí a výkonu, lze jej tedy mimo jiné použít i k dodatečnému testování provozních podmínek motoru. Snímač proudu INA-219 používá ke komunikaci s nadřazeným systémem komunikační sběrnici I2C.

Snímač disponuje šesti registry, jedná se o:

- **Konfigurační registr** - Nastavení rozsahu a rozlišení, na kterém bude snímač měřit
- **Registr úbytku napětí** - Naměřená data o úbytku napětí na měřicím rezistoru
- **Registr zdrojového napětí** - Naměřená data napětí, které napájí zátěž
- **Registr výkonu** - Naměřená data o výkonu na měřené zátěži
- **Registr proudu** - Naměřená data o snímaném proudu
- **Registr kalibrace** - Kalibrace snímače

Na obrázku 3.6 je zobrazeno typické zapojení snímače s jeho vnitřní strukturou. Proud zátěže prochází odporem s označením R_{SHUNT} , na kterém měří úbytek napětí programovatelný operační zesilovač. Signál je poté převeden převodníkem do digitální podoby a následně je zapsán do registrů. Hodnoty registrů se poté odesílají datovou I2C sběrnici do nadřazeného systému.



Obr. 3.6: Schéma vnitřního zapojení snímače INA-219 se zapojením základních elektrických komponent nutných pro funkčnost zařízení.[19]

3.3.4 Výběr interaktivních prvků

Displej

Vzhledem k náročnosti práce s grafikou v embedded systémech byl jako zobrazovací/ovládací prvek zvolen dotykový displej od firmy Nextion. Velikost displeje byla vybrána s ohledem na minimální rozměry zařízení, ale zároveň dostatečnou čitelnost. Z důvodu využití dotykového rozhraní byla zvolena velikost 3.5", z řady *NEXTION ENHANCED HMI DISPLAY* model **NX4832K035**. Firma Nextion se zabývá výrobou displejů primárně sloužících jako průmyslové HMI (Human Machine Interface) panely. Displej se od klasických displejů liší tím, že má vlastní 32-bitový procesor a veškerou elektroniku potřebnou k chodu displeje. Dokáže přes UART přijímat data k zobrazení, popřípadě může odesílat data do nadřazeného systému, která v něm samotném mohou vzniknout interakcí s uživatelem. Díky osazenému procesoru mohou v displeji probíhat krátké, jednoduché kódy zapsané v jazyce vycházejícího z C. Firma Nextion poskytuje vlastní software pro tvorbu grafiky, kde je možné přidávat nebo upravovat indikační či interakční prvky, je možné programovat segmenty kódu, které nastanou při specifické události, a v neposlední řadě debuggovat a nahrávat vytvořenou grafiku do displeje. Displej reaguje na dotyk pomocí odporové dotykové vrstvy, což zajistí, že ani dotyk v rukavicích či v případě vlhkou rukou nebude problém.

Ovládací prvky

I přes možnost používání dotykového displeje v rukavicích, někteří uživatelé upřednostňují haptickou odezvu zařízení na jejich akci. Také v konkurenčních zařízeních je standardem, že krom displeje je zde i několik manuálních ovládacích prvků, jež slouží například k volání základních a často používaných funkcí peristaltické pumpy nebo k pohybu v menu. Tuto funkci bude zajišťovat membránová klávesnice, dvě tlačítka pro zapnutí a vypnutí pumpy a jeden rotační enkodér.

3.3.5 Výběr výkonového řízení

Vzhledem k tomu, že návrh zařízení regulující výkon motoru není předmětem této práce, byl na toto místo vybrán modul H-můstku s integrovaným obvodem **L298N**. Tento před-sestavený modul byl zvolen pro jeho parametry maximálního proudu až 2A a maximálního pracovního napětí až 35V. Tyto parametry umožní zařízení fungovat univerzálně jak pro 12V motory tak pro 24V motory a vzhledem k dualitě H-můstků v čipu L298N je možné připojit i krokový motor, což je další výhoda pro budoucí rozvoj zařízení.

3.4 Ověření komunikace mezi PC a mikroprocesorem

V této kapitole bude vysvětleno a popsáno, jakým způsobem byla testována a zprovozněna komunikace počítače, na kterém běžela aplikace použita jako první klient MQTT protokolu, a mikroprocesoru, na němž byl vytvořen druhý klient.

3.4.1 Tvorba MQTT klienta na mikroprocesoru

MQTT klient na mikroprocesoru je vytvořen za pomoci knihovny *PubSubClient.h*, která zprostředkovává velice přehledné API (Application Programming Interface). Na výpisu 3.1 je ukázka kódu, který připojí mikroprocesor na WiFi a k MQTT serveru *broker.hivemq.com* a začne odebírat téma *peristaltic/test*.

V programu jsou použity dvě typické funkce pro framework *Arduino* ***void setup()***, ve které probíhá nastavení komunikace, připojení se k WiFi a nastavení MQTT serveru, a ***void loop()***, ve kterém se kontroluje, zdali přišla zpráva. Mimo tyto základní funkce je zde i funkce ***void callback()***, která se vykoná vždy, když klient zaznamená zprávu. Zde se konkrétně vypíše téma a zpráva, která přišla.

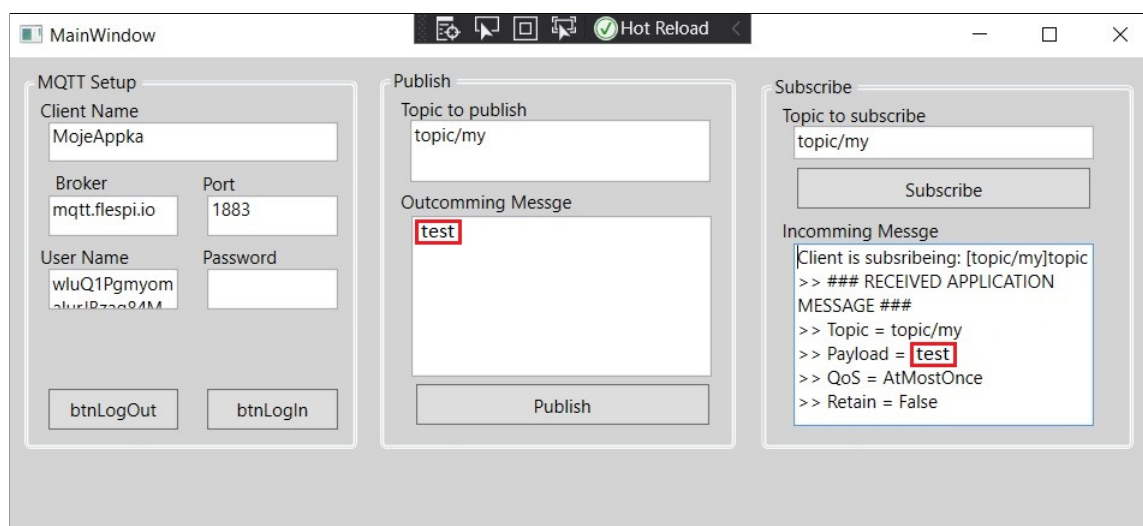
```
1 #include <PubSubClient.h>
2 const char *mqtt_server = "broker.hivemq.com";
3 WiFiClient ethClient;
4 PubSubClient client(ethClient);
5
6 void setup()
7 {
8     Serial.begin(9600);
9     setupWifi();
10    client.setServer(mqtt_server, 1883);
11    client.setCallback(callback);
12    client.subscribe("peristaltic/test");
13 }
14 void loop()
15 {
16     client.loop();
17 }
18 void callback(char *topic, byte *payload, unsigned int length)
19 {
20     Serial.print("Message arrived to: ");
21     Serial.print(topic);
22     Serial.println("] ");
23
24     Serial.print("Message:");
25     for (int i = 0; i < length; i++)
26         Serial.print((char)payload[i]);
27     Serial.println();
28 }
```

Výpis 3.1: Příklad implementace MQTT klienta na mikroprocesoru ESP-32 za pomoci frameworku Arduino a knihovny PubSubClient.h.

3.4.2 Tvorba MQTT klienta na PC

Aplikace, reprezentující klienta pro demonstraci komunikace, byla vytvořena v jazyce C# pomocí frameworku WPF. Tento framework by měl do budoucna zajistit dostatečné grafické možnosti pro reprezentativní vzhled aplikace. Současný stav slouží pouze pro demonstrování backend kódu, který používá knihovnu MQTTnet. Tato knihovna je v programátorské komunitě velmi oblíbená pro svou spolehlivost a jednoduchost používání jejích funkcí. Na obrázku 3.7 je základní grafické rozhraní aplikace testující komunikaci. Aplikace je schopna se připojit k jakémukoliv MQTT serveru, jenž pro přihlášení nevyžaduje certifikát ověření. Všechna nastavení pro změnu serveru, přihlášení a odhlášení jsou v levé sekci aplikace. V prostřední části je místo pro nastavení tématu, kam chceme poslat zprávu (témat může být více najednou). A jako poslední je část pro příjem zpráv, kde lze změnit odebírané téma a lze vidět výpis zpráv v případě, že přijdou.

Pokud v sekci *Subscribe* i *Publish* použijeme stejné téma, můžeme zkontrolovat korektnost komunikace, neboť klient se tak sám sobě stane zdrojem zpráv.



Obr. 3.7: Snímek prvotního návrhu počítačové řídicí aplikace implementující funkce MQTT klienta. Na obrázku je vidět vyznačená odeslaná zpráva a vyznačená přijatá zpráva v terminálu klienta.

3.4.3 MQTT Server

Jako server byl využit online MQTT server *flespi.io*, který byl při vývoji nápomocný s kontrolou komunikace, neboť poskytuje privátní token (přihlašovací údaje). Díky tomu dovoluje monitorovat všechny zařízení připojené k tomuto přihlašovacímu tokenu, může kontrolovat jejich příchozí i odchozí data a zprostředkovává i nástroje "online" klienta pro příjem a odesílání zpráv, které byly využity při testování.

4 Program zařízení

Zdrojový kód zařízení je psán v objektově orientovaném jazyce *C++*, ale vzhledem k implementaci v embedded systému byla k tomuto jazyku použita nadstavba *Arduino*, z důvodu jednoduššího přístupu k ovládání a čtení pinů procesoru, sériové lince a dalších funkcí často používaných v embedded zařízeních. Díky použitému procesoru ESP-32 je možné implementovat i prvky z kategorie IoT, které bude v této kapitole vysvětlen.

4.1 Představení platformy Arduino

Fenomén Arduino odstartoval v roce 2005, kdy byl světu představen jeho první model. Otvíral možnosti domácí automatizace a robotiky široké veřejnosti především svou dostupností a relativně jednoduchým přístupem k programování.

Arduino

Jedná se o jednodeskový mikropočítač s osmibitovou architekturou od společnosti Atmel. Výkon zařízení není srovnatelný například s počítači RaspberryPi, ale je velice schopný, co se týká jednoduchých aplikací. Postupem času s rostoucí komunitou začala růst jeho popularita díky čemuž začal nacházet čím dál tím větší uplatnění. Vzhledem k popularitě zařízení začali ostatní výrobci představovat mikroprocesory s totožnou ideologií. Vzhledem k tomu, že Arduino nebylo schopné dodat dostatek výkonu pro náročnější aplikace, začaly vznikat knihovny, které dovozovaly programovat mikroprocesory ostatních výrobců stejným způsobem.

Platforma Arduino a ESP-32

Ačkoliv platforma Arduino nedovoluje programátorovi tak rozsáhlé možnosti, jako nativní framework ESP-IDF a nepatrně jej zpomaluje, byla pro tento projekt vybrána především pro dostupnost potřebných knihoven, rozsáhlá komunitní fóra a zrychlení vývoje firmwaru pro pumpu.

4.2 Představení fenoménu Internet věcí

V důsledku čím dál tím větší nutnosti vzdáleného monitorování a ovládání zařízení v továrnách, vznikla v roce 2016 premisa informatické infrastruktury propojující fyzická zařízení. Připojená zařízení by díky síťové konektivitě mohla odesílat data ze senzorů či přijímat pokyny k provádění akcí z nadřazeného systému. Ačkoliv každé zařízení dokáže pracovat samostatně, dokáže díky propojení s ostatními zařízeními zvýšit svou autonomnost a vytvořit komplikovanou strukturu, například manufakturní výroby.

IoT (Internet of Things) koncepce je v případě tohoto zařízení skvělým předpokladem nejen ke vzdálenému sběru dat a ovládání, ale i pro možnost zřetěženého dávkování několika pump, které by spolu komunikovaly. Zřetěženým dávkováním by pumpy mohly míchat roztoky z různých kapalin, popřípadě vytvořit dávkovací linku, kde by se druh kapaliny dávkoval na základě požadavku nádoby s čipem či QR kódem.

4.3 Konstrukce programu

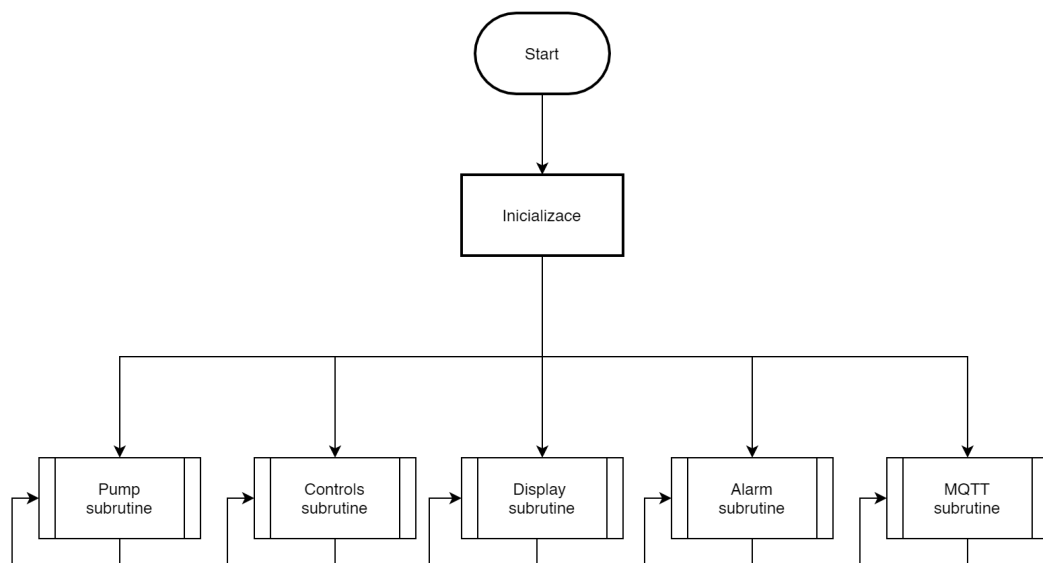
Program byl pro zvýšení přehlednosti rozdělen do pěti subrutin, jmenovitě:

- **Pump** - obsluhuje všechny periférie, které souvisí s pumpou, jako je snímač proudu, driver motoru nebo enkodér
- **Controls** - vyčítá data z tlačítek a rotačního enkodéru, obsahuje také třídu *Menu*, která obsahuje prvky k zobrazení na displeji, v závislosti na vstupních perifériích
- **Display** - obsluhuje displej *Nextion* pomocí sériové linky
- **Alarm** - hlídá mezní hodnoty a vyhodnocuje správný chod pumpy, obsluhuje také LED pásek a bzučák pro vizuální i zvukové upozornění na výjimečný stav
- **MQTT** - zajišťuje komunikaci se serverem, vyčítání a odesílání dat

Každá subrutina má svou vlastní třídu, které sdružuje veškeré členské metody a proměnné důležité k chodu třídy.

Zdrojový kód *main.cpp* obsahuje funkce *void setup()*, ve kterém probíhá inicializace a *void loop()*, který se následně volá cyklicky a zajišťuje běh programu v nekonečné smyčce. Mimo tyto dvě základní třídy jsou zde nadeklarovány i výše zmíněné subrutiny.

Na obrázku 4.1. je vývojový diagram znázorňující zjednodušenou logiku virtuální paralelizace jádra procesoru. Po spuštění se provede inicializační rutina, která nastaví piny, proměnné a komunikaci. Následuje cyklické vykonávání každé ze subrutin.

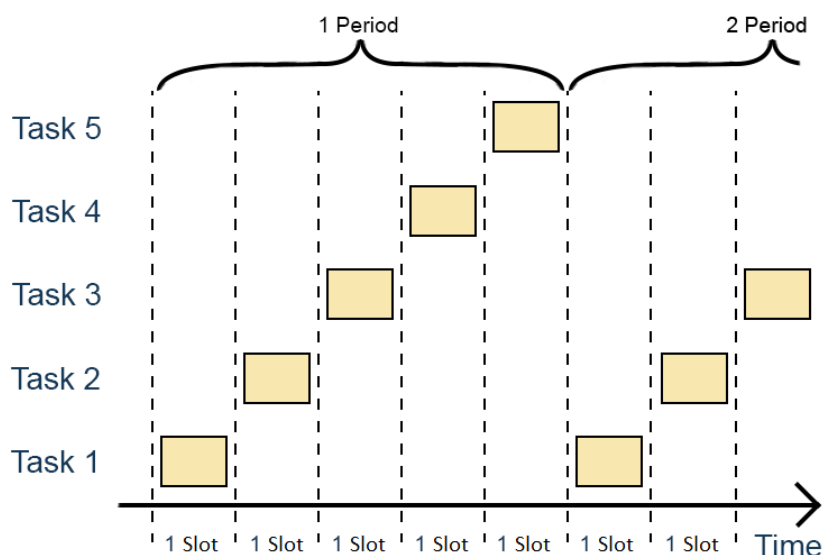


Obr. 4.1: Vývojový diagram znázorňující základní logiku běhu programu. Po prvotním nastavení se začne vykonávat všech pět hlavních podprogramů.

4.4 Virtuální paralelizace - subrutiny

Vzhledem ke komplexnosti programu bylo nutné jej rozdělit do částí, především pro zvýšení přehlednosti kódu a možnosti zapouzdření jednotlivých částí. Komplikace vznikla v bodě, kdy předchozí kód příliš zdržoval části kódu, které bylo nutné vykonávat cyklicky s určitou periodou. Bohužel nebylo možné toto zdržení kompenzovat vzhledem k proměnné délce vykonávání kódu. Řešení tohoto problému nabídla knihovna *AceRoutine.h*, která dovoluje virtuálně "paralelizovat" procesy v jednom jádře procesoru. Princip takovéto paralelizace spočívá v tom, že je kód rozdělen do takzvaných "subrutin", které obsahují jednotlivé programové celky. Tyto subrutiny jsou následně rozděleny takzvaným plánovačem, který rozdělí subrutiny do menších celků, které mohou fungovat samy o sobě. Každý celek kódu má tak vyhrazené časové okno nazvané slot, ve které se vykoná tento úsek kódu z jedné subrutiny. Jakmile je z každé subrutiny proveden jeden kódový celek, cyklus se začne vykonávat od znova a je ukončena jedna perioda cyklu.

Knihovna emuluje subrutiny bez použití zásobníku a nahrazuje vykonávání funkcí jako je *yield()* nebo *delay()* tak, aby nezadržovaly vykonávání kódu a dovolily chod ostatních rutin. [9]



Obr. 4.2: Obrázek ilustruje přepínání subrutin. Každá subrutina má vyhrazené "časové" okno nazvané slot, po které je vykonávána procesorem, než dojde k přepnutí na další.

Knihovna *AceRoutine.h*, která umožňuje takovéto řízení, je optimalizována pro výkon jak po stránce paměti, tak po stránce rychlosti, nicméně vzhledem k tomu, že je tato knihovna nadstavbou, je přístup pomalejší než nativní přístup. Dle dokumentace vývojáře odpovídají časy přepínání na různých platformách orientačně následovně:

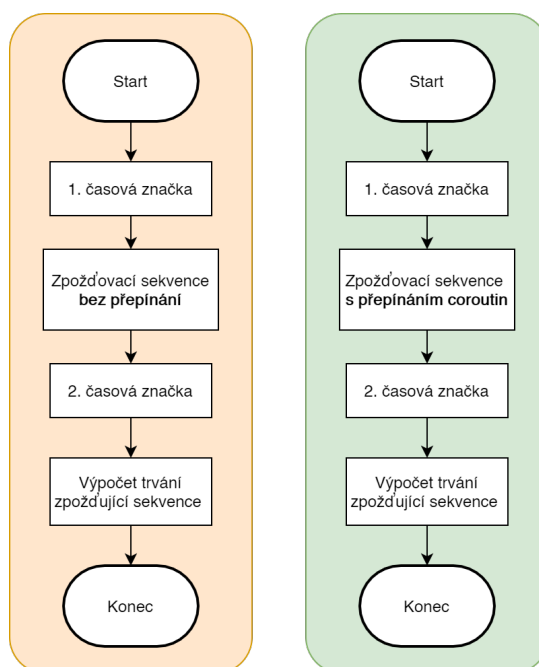
- 5.3 mikrosekund na čipu 16 MHz ATmega328P
- 2.5 mikrosekund na čipu 48 MHz SAMD21
- 1.8 mikrosekund na čipu 72 MHz STM32
- 1.5 mikrosekund na čipu 80 MHz ESP8266
- 0.4 mikrosekund na čipu 240 MHz ESP32

Pro vlastní otestování rozdílu rychlostí mezi použitím funkcí knihovny a použitím nativního prostředí jsem vytvořil program v jazyce C++, který testuje rozdíl času mezi vykonáním stejných funkcí za pomoci volání maker knihovny a přímým voláním funkcí. Vývojový diagram na obrázku 4.3 demonstruje cyklické zapisování jednoho znaku na sériovou linku. Aby byl rozdíl měřitelný, musel být kód vykonán několikrát po sobě.

Tabulka 4.1 demonstruje rozdíly časů v případě využití maker subrutin a nativního přístupu. Bylo provedeno několik měření s různým počtem cyklů. V případě využití přepínání cyklů bylo dle očekávání trvání zpožďovací sekvence delší, než v případě nativního volání vypisování. Rozdíl času ku počtu cyklů byl však ve všech případech téměř totožný s časem udávaným vývojáři knihovny *AceRoutine.h*.

Počet cyklů [cykly]	Bez přepínání [us]	S přepínáním subrutin [us]	Rozdíl časů [us]	Čas jednoho přepnutí [us]
5	68	71	3	0,60
10	131	135	4	0,40
20	258	265	7	0,35
30	395	409	14	0,47

Tab. 4.1: Tabulka zobrazuje rozdíl času běhu zpožďovacích sekvencí za využití standardního volání a přepínáním korutin. Pro zvýšení rozlišitelnosti měření bylo měření provedeno při vícenásobném počtu přepnutí.



Obr. 4.3: Vývojové diagramy programu měřícího rozdíl času zpožďovacích sekvencí za využití nativního volání a za využití přepínání s pomocí subrutin.

4.5 Třídy programu

Vzhledem k objektově orientovanému programování byl celý projekt rozdělen do tříd, kdy každá z nich má na starost jednu z pěti hlavních oblastí programu. V této kapitole budou rozebrány a vysvětleny všechny třídy a jejich hlavní funkce. Pro použití instancí tříd ve zdrojovém kódu má každá instance hlavní API (Application Programming Interface) nazvané *loop()*, které vstoupí do třídy a obstará volání všech potřebných metod pro splnění účelu instance.

4.5.1 Třída Pump

Tato třída sdružuje všechny informace o pumpě, o jejím nastavení, stavu a parametrech. Obsahuje řídicí algoritmy, které budou podrobně rozebrány v kapitole 5.4.

Jsou zde deklarovány instance využívané v této třídě. Jedná se o:

- měřicí modul INA219
- magnetický enkodér na hřídeli motoru
- výkonový řídicí H-můstek

V této třídě jsou také definovány a použity následující pomocné třídy:

- **Settings** - třída obsahující neměnné parametry pumpy
- **Parameters** - třída sdružující veškeré proměnné parametry k běhu
- **Sensor** - ukládá data ze senzoru proudu
- **Status** - proměnné v této třídě udržují informace o aktuálním stavu pumpy

Mimo hlavní metody povolující chod pumpy jsou zde metody na vyčítání stavů pumpy, metody pro úpravu nastavení a metoda *void settings()*, která upraví globální nastavení pumpy po příjmu zprávy z MQTT serveru.

Třída samotná obsahuje také instanci třídy *Adafruit INA219*, která zajišťuje komunikaci s čipem INA219 a zprostředkovává data z ní, a instanci *ESP32Encoder*, která informuje o otáčkách enkodéru motoru.

4.5.2 Třída Controls

Veškeré mechanické ovládací prvky, jako jsou tlačítka a rotační enkodér, jsou obsluhovány v této třídě. Zahrnuje tři instance instance třídy *Button2* pro tlačítka enkodéru, START a STOP tlačítko. Tato knihovna byla zvolena, protože má předpřipravené API pro různé události tlačítka jako je stisknutí, puštění, dlouhý stisk a především ošetření zákmitů. Mimo to se zde definují piny, na kterých jsou všechny interakční prvky připojeny.

Kromě těchto tlačítek třída obsluhuje také vyčítání z ovládacího enkodéru. V hlavní metodě *loop()* se zjišťuje, zdali nastala událost na tlačítkách a reaguje na ně.

4.5.3 Třída MQTT

Tato třída zajišťuje připojení se k WiFi a MQTT serveru. Zprostředkovává vyčítání ze serveru, přihlašování se k odebrání témat, odesílání zpráv a kontroluje, jestli je zařízení stále připojeno. Pokud se zařízení nečekaně odpojí, upozorní uživatele a pokusí se znovu připojit.

4.5.4 Třída Display

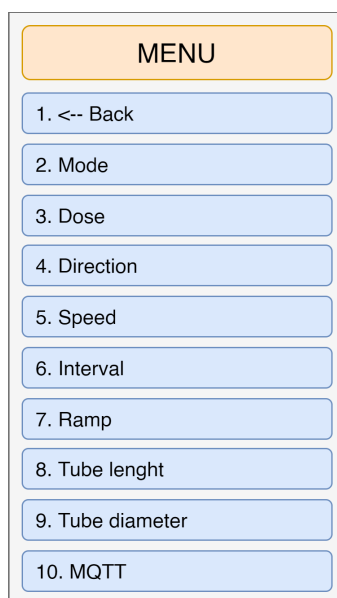
Třída *Display* obsluhuje komunikaci s displejem *Nextion*. Nese informace o aktuálně zapsaných údajích na displeji a jeho nastavení. Obsluhuje metody pro zápis do konkrétních prvků displeje, přepínání stránek a metody vyvolané dotykem na displeji.

Menu

Třída *Display* mimo jiné obsahuje i instanci třídy *Menu*, která zajišťuje zobrazení a pohyb v položkách nastavení. Zprostředkovává tak uživateli přístup ke všem parametrům pumpy v přehledném rozpoložení viz. obrázek 4.4.

Hlavními metodami jsou zde metody pro:

- pohyb v menu
- výběr prvku
- ovlivňování hodnoty vybraného prvku
- vkládání změněné hodnoty do nastavení pumpy



Obr. 4.4: Seznam prvků menu pumpy. Mezi prvky se lze libovolně pohybovat a měnit jejich konstanty. Zobrazené prvky jsou vždy jen čtyři najednou

4.5.5 Třída Alarm

Zařízení měří kontinuálně chybové stavy pumpy a pokud nějaký z nich nastane, signalizuje je pumpa vizuálně i zvukově. Obsluhu těchto dvou komponent má na starosti právě třída *Alarm*. Třída obsahuje nastavení, co se týká signalizačních pinů pro LED pásek a bzučák. Obsahuje jejich nastavení, které se mění v závislosti na úrovni dosaženého alarmu. Její hlavní proměnnou je instance třídy *Freenove_ESP32_WS2812* nazvaná *Strip*. Tato instance zajišťuje ovládání LED pásku adresovatelných LED diod typu WS2812, poskytující širokou barevnou škálu pro různé úrovně událostí, na které může vizuálně upozornit.

4.6 Algoritmy řízení

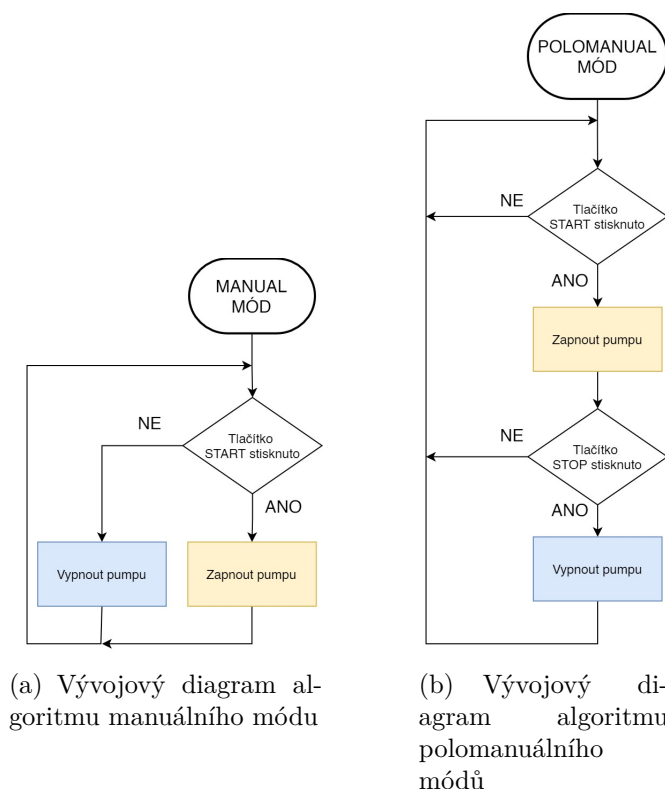
Pro možnosti víceúčelového využití pumpy byly vytvořeny čtyři řídicí módy, ve kterých je pumpa schopna operovat. Módy mění především přístup k chodu pumpy a podle toho se rozlišují na módy vyžadující lidskou kontrolu a módy, na základě jejichž požadavků požadavků od uživatele dokáže pumpa sama nadávkovat požadované množství kapaliny. Módy pumpy lze měnit buď na klávesnici na čelním panelu pumpy, nebo na obrazovce zařízením dotykem klepnutím na indikátor módu, nebo na obrazovce *nastavení* rotačním enkodérem.

4.6.1 Plně manuální mód

Mód manuálního řízení zajišťuje základní druh ovládání pumpy. Pumpa v tomto módu běží pouze v případě, drží-li uživatel tlačítko START. Jakmile je tlačítko puštěno, pumpa se zastaví. Princip řízení tohoto módu ukazuje obrázek 5.4. (a).

4.6.2 Polo-manuální mód

Mód je vhodný pro dávkování většího množství objemu s nutností kontroly uživatele. Jedná se o další základní druh řízení, který odstraňuje nutnost držení tlačítka START pro kontinuální chod pumpy. Pokud se stiskne tlačítko START, pumpa je uvedena do chodu a běží do té doby, dokud není stisknuto tlačítko STOP. Styl řízení tímto algoritmem je ilustrován na obrázku 5.4 (b)



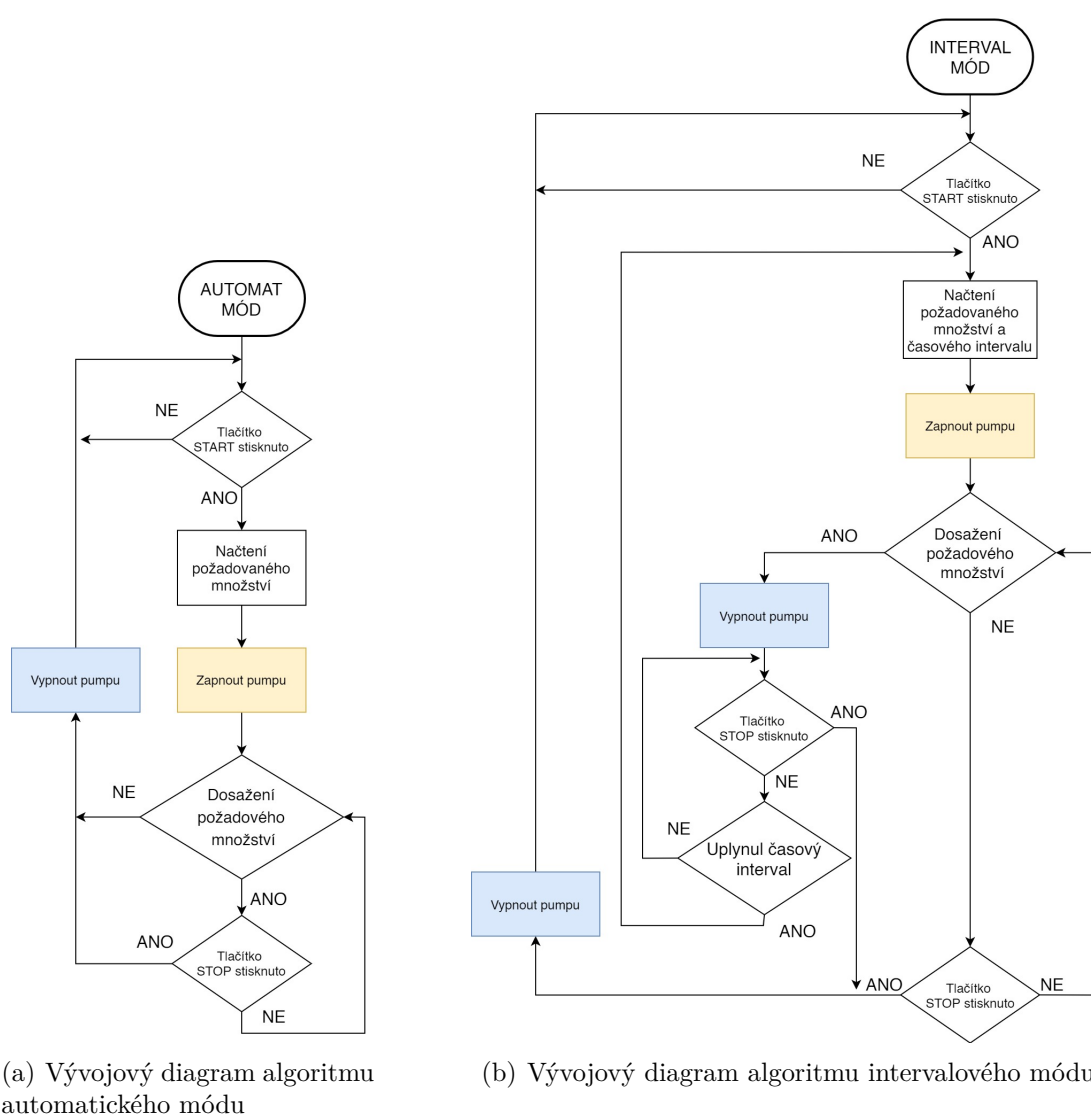
Obr. 4.5: Vývojové diagramy základních módu pumpy

4.6.3 Automatický mód

Automatický mód dělá z pumpy velmi přesný dávkovač. Tento mód využívá modulu magnetického snímače otáček na motoru pumpy. Díky vlastnosti snímače předat 12 pulzů na jednu otáčku a vysokému převodu mezi hřídelí motoru a samotnou hřídelí hlavy pumpy, získáváme předpoklad k velmi vysoké přesnosti monitorování přenesených mililitrů. Pumpa v tomto módu odměří požadované množství mililitrů a poté se vypne.

4.6.4 Intervalový mód

Funguje stejně jako výše zmíněný automatický mód s tím rozdílem, že po přenesení požadovaného množství se řízení nevypne, ale pouze pozastaví na dobu nastavenou uživatelem. Po uplynutí intervalu se pumpa opět spustí a přenese další dávku.



Obr. 4.6: Vývojové diagramy pokročilých řídicích módů pumpy

5 Ovládací aplikace

Ovládací aplikace je doplňkový software, který byl vyvinut pro rozšíření funkčnosti pumpy, a jeho primární podstatou bylo zaznamenávání měřených údajů z pumpy, její vzdálené ovládání a měnění nastavení a parametrů. Pumpa tedy může být plně monitorována i ovládána bez nutnosti přítomnosti operátora.

5.1 Účel a funkce aplikace

Aplikace slouží jako vzdálený ovladač a monitor pumpy, se kterou komunikuje pomocí protokolu MQTT, jenž byl podrobně rozebrán v kapitole 2.1 MQTT komunikace. Prvotní přístup k pumpě byl zvolen bezdrátově vzhledem k IoT koncepci zařízení, nicméně drátový přístup je plánovaným rozšířením stávajícího zařízení především z důvodu spolehlivosti. V ideálních podmínkách, jako je spolehlivý a silný přístup k bezdrátovému připojení k internetu, jsou však rozdíly mezi drátovým a bezdrátovým přístupem stírány vzhledem k nárokům na rychlost komunikace a množství přenesených dat.

Primárním účelem aplikace bylo monitorování veličin pumpy. Co se monitorování týká, dokáže sledovat následující:

- Chod motoru
- Směr otáčení
- Přenesené množství média
- Proud do motoru v čase
- Aktuální nastavení
- Chybová hlášení

Mimo monitorování je skrze aplikaci také možné celé zařízení plnohodnotně ovládat. Hlavní tlačítka START a STOP jsou v nejvyšší vrstvě uživatelského rozhraní, takže nikdy nebudou skryta, což uživateli zajišťuje rychlý a nekomplikovaný přístup k této elementární funkcionalitě aplikace. Z hlavní monitorovací stránky se lze jednoduše prokliknout na stránku nastavení, kde jsou zobrazena všechna dostupná nastavení pumpy, které uživatel po změně potvrdí a odešle do pumpy. Pro účely testování a ověření funkčnosti spojení s pumpou jsem do aplikace zabudoval mimo samotné implementace MQTT komunikace i celý klientský MQTT terminál s uživatelským rozhraním v souladu s grafickou koncepcí aplikace. Terminál dovoluje připojit se na jakýkoliv hostující MQTT server a fungovat mimo primární účel aplikace. Pokud je ale přepnuto nastavení na této stránce, platí automaticky pro celou aplikaci, což způsobí ztrátu komunikace s pumpou

5.2 Problematika návrhu uživatelského rozhraní

UX (User Experience) a UI (User Interface) jsou zkratky v překladu vyjadřující uživatelskou zkušenost s návrhem a uživatelské rozhraní. UX je část návrhu pojednávající o funkčnosti designu a UI pojednává o způsobu, jak s ním uživatel interaguje. V případě této práce se však jedná zejména o grafické uživatelské rozhraní čili GUI.

Uživatelské rozhraní

Tato část návrhu designu se zabývá zcela výhradně vizuální stránkou návrhu, nikoli jeho funkčností. Jejím cílem je uživatele na první pohled zaujmout a udržet jeho pozornost. Ač to může působit v rozporu s předchozím tvrzením, design by neměl uživatele rušit od jeho cílů a úkolů, které chce plnit, a měl by být téměř neviditelný. Návrh by měl být především líbivý, v závislosti na druhu aplikace hravý nebo naopak bez rušivých elementů profesionálně strohý. Do návrhu by se měla promítnout spojitost s firmou či produktem, který design používá.

Uživatelská zkušenost

Jedná se o velice rozsáhlou kategorii návrhu designu, která navzdory tomu, že ač je na první pohled neviditelná, je považována za mnohem důležitější než samotný grafický návrh. Jedná se především o psychologický přístup k ovládání aplikace, jehož cílem je uživateli co nejvíce usnadnit ovládání, či jej podvědomě navést k cíli. Pracuje se zde především s:

- rozpořádáním ovládacích a informačních prvků do mřížky
- tvorbou schémat
- rozvržením velikostí objektů dle jejich důležitosti
- uzpůsobením výraznosti barev dle důležitosti objektů
- volbou fontů

Správný UX design je jednoduchý, samovysvětlující a responsivní. Design tak dokáže uživatele i na první pohled provést celou aplikací, aniž by byl uživatel zmaten nebo ztracen. Tato oblast tvorby designu se zabývá spíše psychologii a uvažováním uživatelů, a proto je pro kvalitní návrh nezbytné znát cílovou skupinu, a v průběhu návrhy testovat na uživateli a reagovat na jejich podněty.

5.3 Představení frameworku Windows Presentation Foundation

WPF (Windows Presentation Foundation) je framework pro uživatelské rozhraní desktopových aplikací. Důvodem k výběru tohoto frameworku byla široká komunitní základna. Využívaný programovací jazyk je C#, který vychází z ideologie programovacích jazyků C. Pro tvorbu grafického rozhraní je defaultně zvolen jazyk XAML. Nevýhodou tohoto frameworku je bezesporu nemultiplatformnost, neboť může běžet pouze pod operačním systémem Windows. Návrhové vzory jsou však podobné frameworku Xamarin, takže například přenesení aplikace na mobilní zařízení by nemusel být tak velký problém.

Programovací jazyk C#

C # je objektově orientovaný vysokoúrovňový jazyk vycházející z jazyka C++ a Java. Mezi jeho přednosti patří především jednoduchost a univerzálnost. I přes svůj

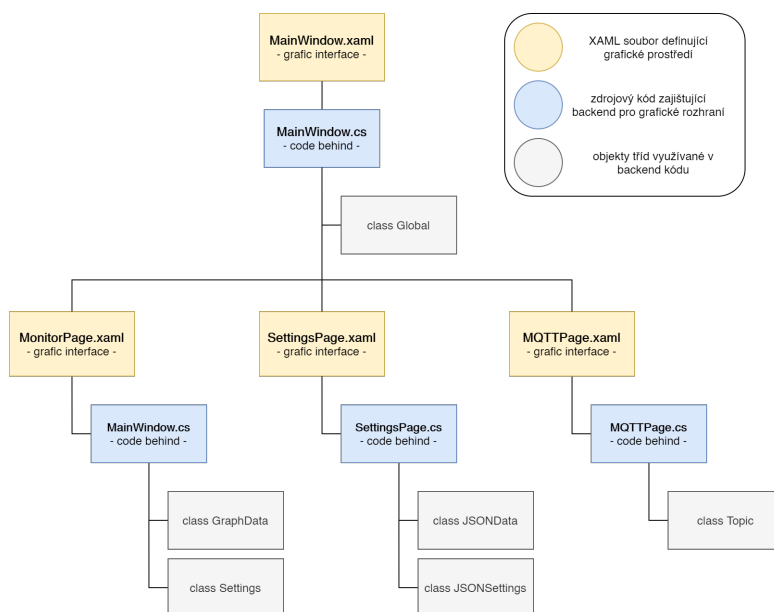
status vysokoúrovňového jazyka je jazyk navržen tak, aby neplýtl časem a pamětí procesoru, a proto se projekty napsané v tomto jazyce řadí k těm rychlejším. Nemůže však konkurovat jazykům jako je C, Assembler nebo Rust. Jazyk má mnoho funkcí podporujících zrychlení vývoje, jako je detekci hranic polí, typová bezpečnost nebo algoritmus pro automatickou správu paměti, takzvaný garbage collector. Projekty psané v tomto jazyce jsou robustní, přehledné a lehce udržitelné.

Programovací jazyk XAML

XAML (Extensible Application Markup Language) je deklarativní jazyk vytvářející instance objektů, kterým dokáže velmi jednoduše měnit vlastnosti. Lze zde velmi jednoduše vytvářet mřížky, bloky nebo načítat data z běžícího programu. Obecně má velmi příjemné prostředky pro tvorbu moderního UI s kvalitním UX. Krom vizualizace při přípravě .xaml souboru lze využít vlastnosti vykonávání vykreslování prvků při běhu programu, což umožňuje designovat prvky v průběhu spuštění aplikace, kontrolovat vzhled jednotlivých komponent a vyvarovat se tak kolizi s jinými prvky.

5.4 Architektura aplikace

Jak bylo zmíněno výše, aplikace byla vytvořena ve frameworku WPF, čemuž byla přizpůsobena stavba aplikace. Schéma rozložení aplikace je znázorněno na vývojovém diagramu na obrázku 5.1. Aplikace je rozdělena do čtyř oblastí dle jejich účelu. Každá má na starost jiný aspekt, který byl zapouzdřen tak, aby byla aplikace přehledná a udržitelná pro budoucí vývoj.



Obr. 5.1: Diagram vyjadřující architekturu aplikace. Architektura zohledňuje strukturu WPF projektů. Nadřazenou třídou je MainWindow, která v sobě sdružuje zbylé tři třídy MonitorPage, SettingsPage a MQTTPage

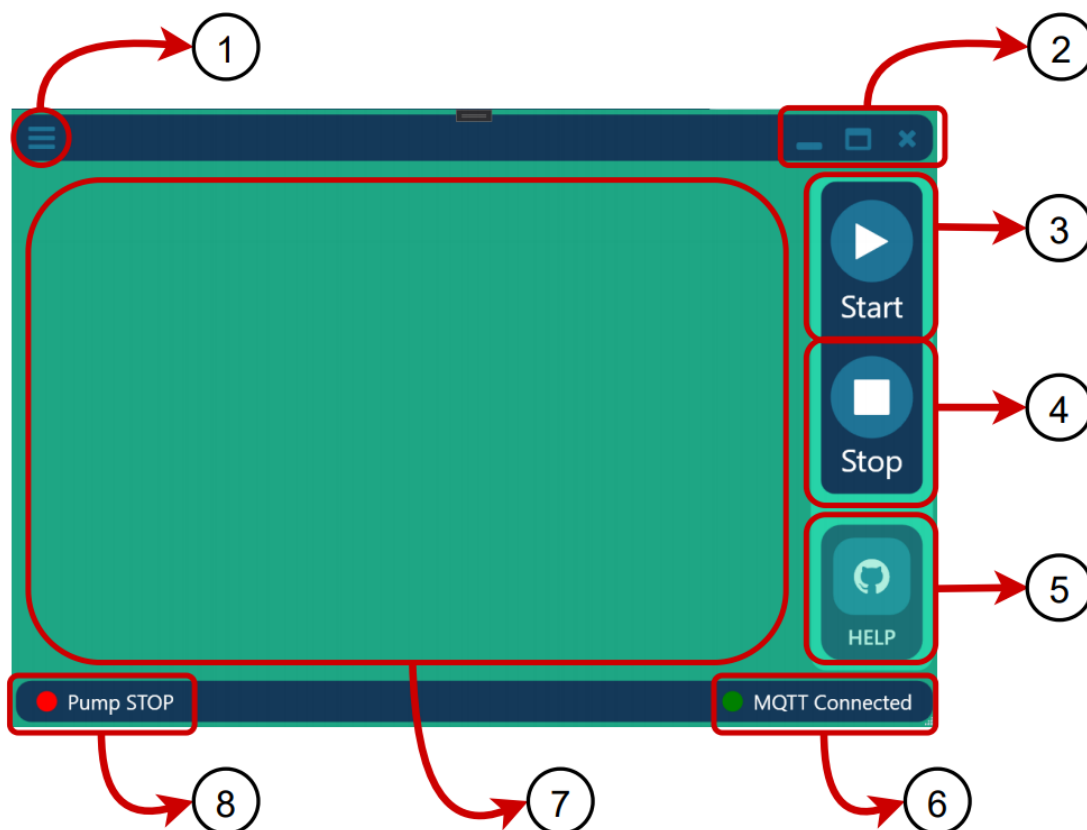
5.4.1 MainWindow

Tato vrstva aplikace sdružuje všechny ostatní, protože má charakter okna. Snímek vrstvy se i s popisem nachází na obrázku 5.2. Obsahuje ikony a funkční tlačítka, které by měly být přístupná z každé části aplikace. Jedná se zejména o:

- systémové funkce minimalizace, maximalizace a vypnutí aplikace
- hlavní navigace v aplikaci
- tlačítka pro zapnutí a vypnutí pumpy
- stavovou lištu informující o stavu pumpy a připojení k pumpě
- tlačítko nápovědy

Popis

1. **Primární rozcestník** - po kliknutí rozbalí lištu pro přechod mezi stránkami
2. **Systémové funkce** - ikona minimalizace, maximalizace a ukončení aplikace
3. **Tlačítko *START*** - po kliknutí vyšle příkaz k zapnutí pumpy
4. **Tlačítko *STOP*** - po kliknutí vyšle příkaz k zastavení pumpy
5. **Tlačítko *HELP*** - po kliknutí otevře v prohlížeči git repositář se souborem README.MD
6. **Indikátor připojení k serveru** - zobrazuje stav připojení k serveru, popřípadě určí druh chyby selhání připojení
7. **Indikátor běhu pumpy** - indikuje RUN/STOP stav pumpy



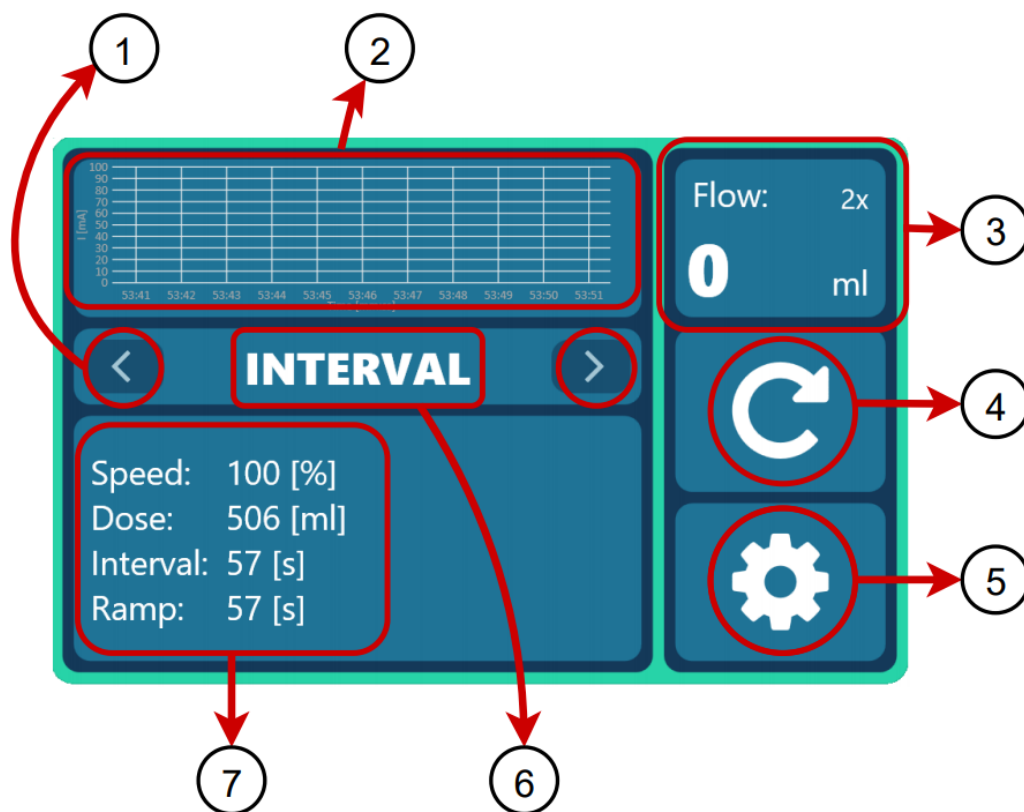
Obr. 5.2: Hlavní okno aplikace sdružující všechny následující vrstvy aplikace. Na obrázku jsou vyznačeny všechny aktivní i indikační prvky.

5.4.2 MonitorPage

Vrstva *MonitorPage* má stejně jako zbytek vrstev charakter stránky, díky čemuž může být zobrazována v objektu *Frame*, který je součástí nadřazené vrstvy. Tato stránka je výchozí stránkou při spuštění. Stránka kopíruje vzhled výchozí stránky displeje na zařízení pumpy pro lepší orientaci uživatele. Popis vrstvy je na obrázku 5.3.

Popis

1. **Tlačítka pro změnu módu** - kliknutím se změní mód chodu pumpy a odešle se do pumpy, nelze měnit při chodu
2. **Graf průběhu proudu** - indikační prvek zobrazující vývoj proudu do motoru pumpy v čase
3. **Indikátor přeneseného objemu** - indikační prvek zobrazující množství, které pumpa přenesla, v případě intervalového módu zobrazuje i počet dávek, kliknutím se počítadlo vynuluje
4. **Indikátor běhu a směru chodu** - indikační ikona, běží-li pumpa, rotuje ve směru chodu pumpy, pokud pumpa neběží, lze kliknutím změnit chod
5. **Tlačítko nastavení** - kliknutím se zobrazí stránka s nastavením
6. **Indikátoru módu** - zobrazuje, v jakém módu pumpa aktuálně pracuje
7. **Informační panel** - při standardním stavu zobrazuje důležitá nastavení pro aktuální mód, při výjimečném stavu zobrazí podrobnosti o chybě



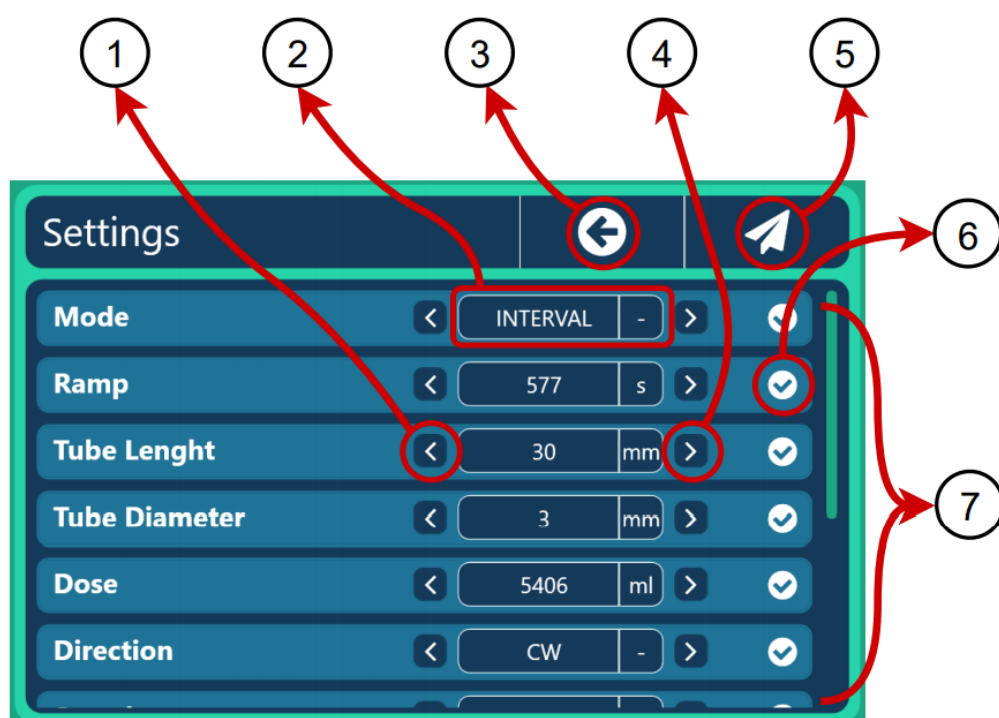
Obr. 5.3: Stránka poskytující změnu základních parametrů jako je mód řízení a směr chodu pumpy. Především ale poskytuje uživateli informace o aktuálním stavu pumpy.

5.4.3 SettingsPage

Ve vrstvě *SettingsPage* se nachází seznam všech uživatelem ovlivnitelných parametrů pumpy. Stejně jako stránka pro monitorování, kopíruje stránka pro nastavení vzhled stránky pro nastavení na displeji pumpy. Většina parametrů je možno zadat číselně přímo do textového pole, ale specifické parametry nelze vyjádřit číselně, a proto bylo přidáno inkrementační a dekrementační tlačítko pro jednodušší úpravu těchto parametrů. Po úpravě parametrů je nutné změny potvrdit a odeslat do pumpy. Vrstvu nastavení lze vidět i s popisky na obrázku 5.4.

Popis

1. **Dekrementační tlačítko** - kliknutím sníží hodnotu parametru o 1, pokud je drženo, mění hodnotu rychle
2. **Textové pole hodnoty parametru** - u číselných parametrů lze zapsat hodnotu přímo z klávesnice, zobrazuje aktuální hodnotu parametru
3. **Tlačítko zpět** - kliknutím odkáže na stránku pro monitorování
4. **Inkrementační tlačítko** - kliknutím zvýší hodnotu parametru o 1, pokud je drženo, mění hodnotu rychle
5. **Tlačítko odeslání nastavení** - kliknutím se aktuální nastavení potvrdí, uloží a odešlou do pumpy
6. **Indikátor synchronizace parametru** - indikuje, zdali aktuální hodnota parametru odpovídá hodnotě v pumpě
7. **Seznam parametrů** - seznam všech dostupných parametrů



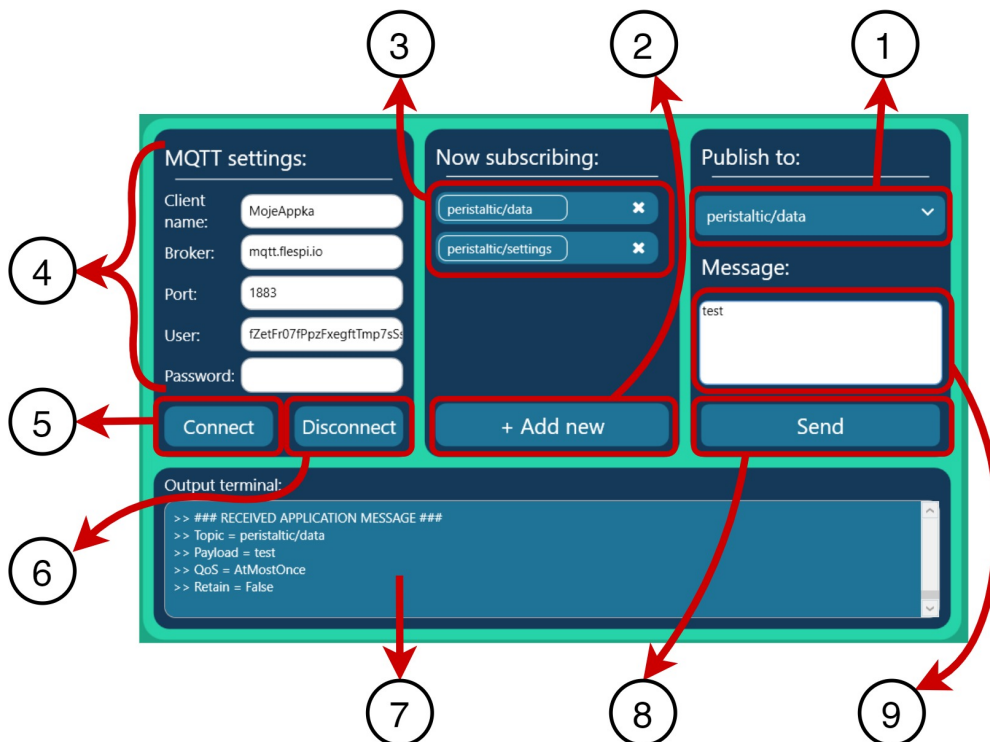
Obr. 5.4: Stránka aplikace sdružující všechny uživatelem ovlivnitelné parametry pumpy. Nachází se zde tlačítka pro úpravu konkrétních parametru, odeslání nastavení do pumpy a vracení se zpět na monitorovací stránku.

5.4.4 MQTTPage

Do této vrstvy je možné se dostat pouze přes primární rozcestník a před uživatelem je nepřímo skryta. Tuto vrstvu uživatel nepotřebuje pro hlavní účel aplikace, nicméně se v ní dá kontrolovat veškerá komunikace se serverem. Stránka *MQTTPage* tvoří plnohodnotné rozhraní MQTT klienta a je pozůstatkem z prvotní fáze projektu, kdy bylo nutné ověřit komunikaci mezi procesorem a aplikací. Všechny prvky stránky jsou zobrazeny na obrázku 5.5.

Popis

1. **Výběr tématu** - kliknutí rozbalí nabídku dostupných témat k publikaci
2. **Tlačítko přidání nového tématu** - kliknutím se objeví formulář pro zadání nového tématu k odebírání
3. **Indikátor právě odebíraných témat** - zobrazuje seznam všech témat, které klient právě odebírá
4. **Formulář parametrů pro připojení** - sdružuje textová pole obsahující parametry pro připojení k MQTT serveru
5. **Tlačítko *CONNECT*** - kliknutím se dle výše zadaných parametrů (4) klient pokusí připojit k serveru
6. **Tlačítko *DISCONNECT*** - kliknutím se klient odpojí od serveru
7. **Terminál klienta** - zobrazuje veškerou komunikaci mezi klientem a serverem
8. **Tlačítko odeslání zprávy** - odešle textový řetězec z textového pole (9) do vybraného topicu (1)
9. **Textové pole pro zprávu** - pole pro text k odeslání na server

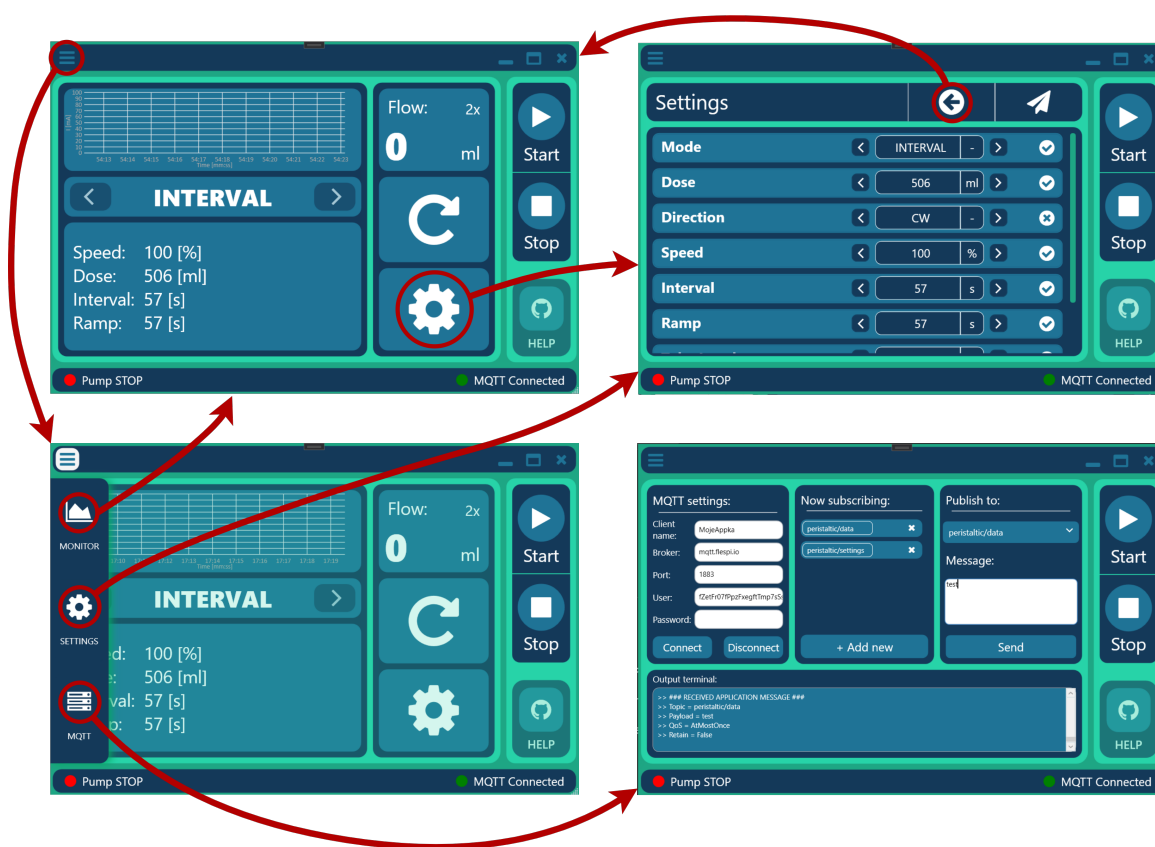


Obr. 5.5: Stránka sdružující všechny interaktivní prvky pro plnohodnotné využívání služeb klienta. Jedná se o servisní stránku, ke které nemá uživatel standardně přístup.

5.4.5 Pohyb v aplikaci

Pro pohyb v aplikaci mezi všemi stránkami může být využit primární rozcestník, který je vždy dostupný v každé části aplikace. Tento rozcestník obsahuje 3 tlačítka s piktogramem a popisem stránky, na kterou odkazuje. Vzhledem k tomu, že v 90% případů se bude uživatel pohybovat pouze mezi stránkou monitorování a stránkou nastavování parametrů, je pro něj mnohem pohodlnější využívat mimo rozcestník tlačítko *Nastavení* na monitorovací stránce pro rychlý přechod na stránku nastavení a tlačítko *Zpět* na stránce *Nastavení* pro návrat na monitorovací stránku. Díky rozcestníku se však lze dostat i do stránky *MQTTPage*. Z této stránky se lze dostat pouze za pomoci rozcestníku.

Veškerý pohyb v aplikaci je graficky prezentován na obrázku 5.6.



Obr. 5.6: Popis pohybu v ovládací aplikaci. Kolečko označuje oblast kliknutí, šipka ukazuje, jak se změní prostředí po kliknutí.

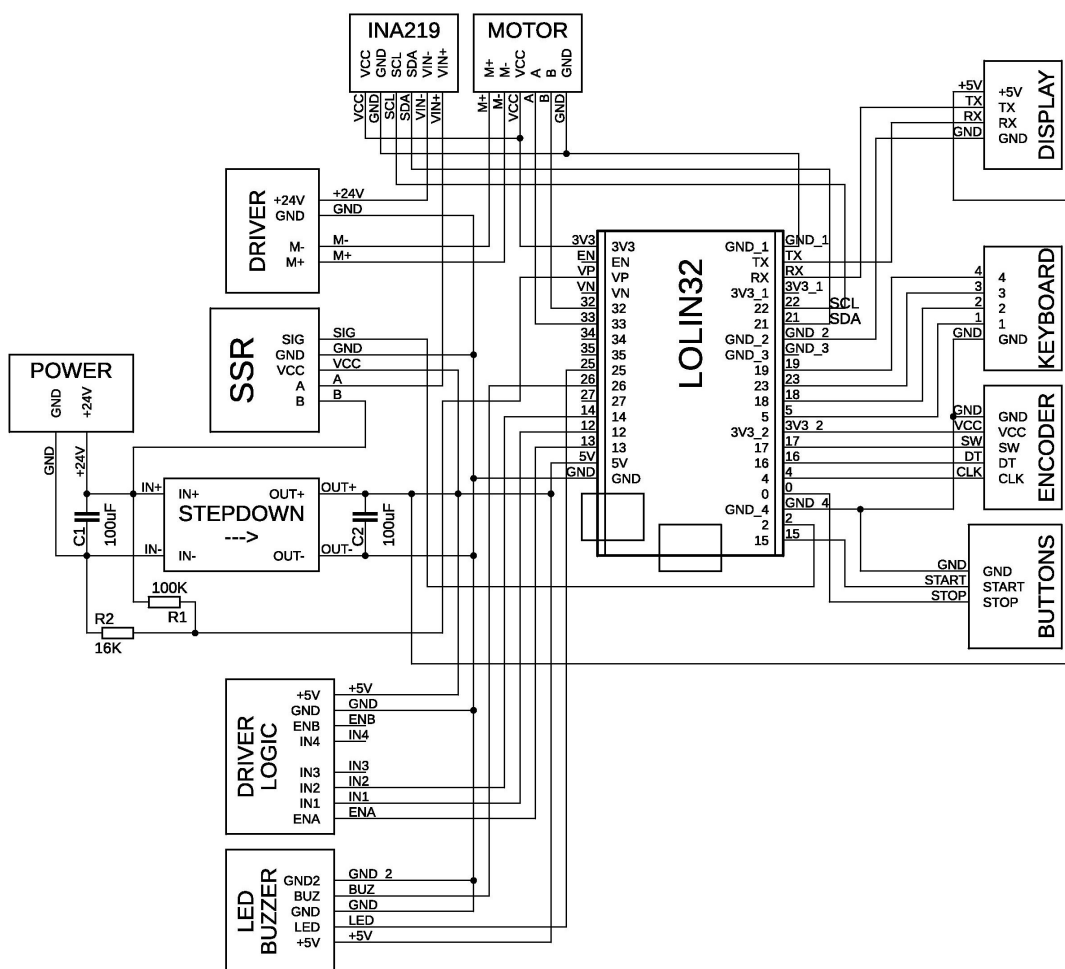
6 Realizace zařízení

V této kapitole vysvětlím dílčí části návrhu finálního zařízení, jejich návrh, realizaci a výsledné propojení dílčích celků. Především v krátkosti popíšu návrh desky plošných spojů a 3D modelu pumpy.

6.1 Deska plošných spojů

DPS (Deska plošných spojů) byla navržena za pomoci softwaru Eagle od společnosti Autodesk. Program Eagle obsahuje schématický editor pro tvorbu schémat a editor tvorby DPS. Výhodou je provázání těchto dvou sekcí, což usnadňuje úpravy schématu a pomáhá vyhnout se chybám.

Deska samotná byla koncipována jako propojovací deska pro zbytek modulů. Nachází se na ní pouze konektory pro propojovací kabely, a patice pro procesor a modul měřící proud. Tento přístup byl zvolen především pro zjednodušení prototypování a eliminaci chyb v návrhu.



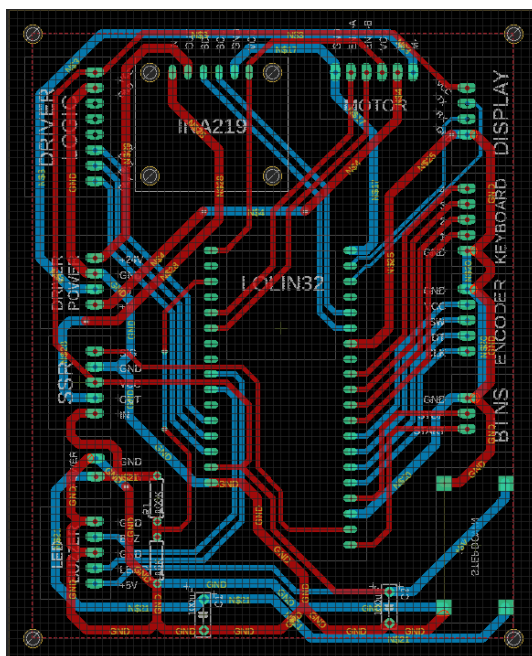
Obr. 6.1: Schéma reprezentující elektrické zapojení zařízení.

Na obrázku 6.2 je znázorněné celé propojovací schéma obvodu. Pro každou součástku byl vytvořen speciální symbol bloku s piny. Součástky jsou reprezentovány formou bloků, jednak z důvodu zvýšení přehlednosti schématu, a jednak protože samotné bloky jsou ve skutečnosti pouze konektory pro propojovací kabely k součástkám. Umístění součástek na schématu částečně koresponduje s rozložením na DPS. Bylo tím zajištěno, že při návrhu budou součástky co nejbližže pinům, na kterých budou reálně připojeny na DPS.

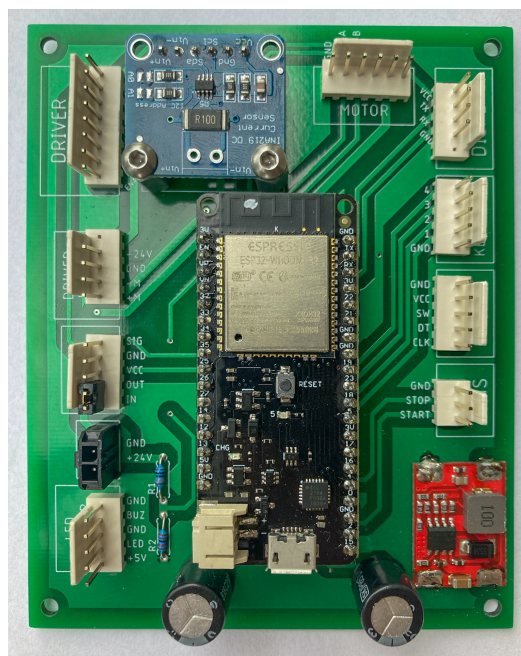
Při tvorbě schématu bylo použito několik tlouštěk měděných cest pro napájení motoru, napájení pro řídicí elektroniku, signály a zem. Všechny cesty byly sice značně naddimenzovány z pohledu jejich maximální zátěže, ale jejich rozdělení pomáhalo při návrhu plošného spoje. Vzhledem k velikosti plošného spoje nebylo nutné uzpůsobit šířku cest jejich předpokládanému maximálnímu proudu. Aktuální zatížitelnost cest je rozepsána v tabulce Tab 6.1. Hodnoty jsou vztahovány k podmínkám cest na povrchu desky za teploty okolí při 20°C.

Typ cesty	Šířka [mm]	I_{MAX} [A]
Napájení motoru	1,4	3,8
Napájení řídicí elektroniky	0,75	2,5
Signální cesty	0,5	1,7

Tab. 6.1: Zatížitelnost všech typů cest na desce plošných spojů



(a) Návrh desky plošných spojů v editoru desek v aplikaci Eagle



(b) Reálná osazená deska

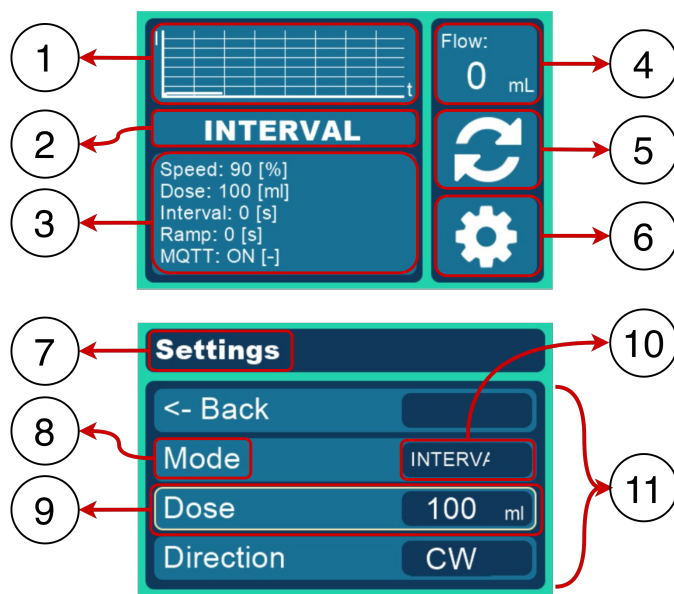
Obr. 6.2: Proces návrhu desky plošných spojů.

6.2 Grafické rozhraní zařízení

Za pomoci softwaru od firmy Nextion bylo vytvořeno grafické a dotykové rozhraní pro dotykový displej Nextion. Grafika, barvy i ovládání byly voleny tak, aby korespondovaly s grafickým rozhraním, aplikace pro uživatele se tak vše jevilo jednotně. Ušetří se tak čas, který by byl vynaložen pro prozkoumávání nového rozhraní. Na obrázku 6.3 jsou zachyceny obě stránky uživatelského rozhraní zařízení s popisky jednotlivých elementů.

Popis

1. **Graf** - indikace průběhu proudu do pumpy
2. **Indikace aktivního módu** - zobrazuje mód, který má pumpa aktuálně nastavený, dotykem lze mód změnit
3. **Informační panel** - ve standardním režimu zobrazuje nastavení, které je pro aktuální mód důležité, signalizuje poruchové stavy a zprávy uživateli
4. **Počítadlo mililitrů** - zobrazuje počet přenesených mililitrů, dotykem je počítadlo vynulováno
5. **Směr otáčení** - směr šipek zobrazuje, jakým směrem se pumpa bude při běhu otáčet, ikona při běhu rotuje, dotykem lze směr změnit
6. **Ikona nastavení** - dotyk zobrazí stránku s nastavením
7. **Název stránky** - informuje uživatele, na jaké stránce se nachází
8. **Název prvku menu** - textové pole zobrazující název prvku v menu
9. **Vybraný prvek v menu** - vybraný prvek v menu indikuje změna barvy jeho rámečku, kliknutím na enkodér změni barvu v pozadí a umožní změnu hodnoty
10. **Hodnota prvku menu** - zobrazuje aktuální hodnotu nastavení, při vybrání prvku a rotaci enkodéru se hodnota mění
11. **Prvky menu** - sekce zobrazující aktuální zobrazované prvky menu



Obr. 6.3: První a druhá stránka grafického rozhraní zařízení pumpy s popisem jednotlivých prvků.

6.3 Návrh a výroba šasi

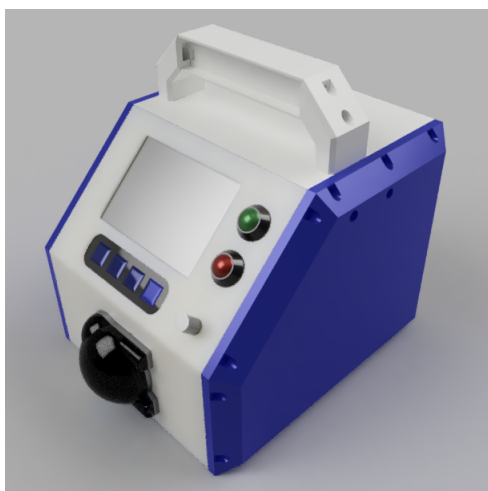
Pro výrobu šasi zařízení byla použita technologie aditivního přidávání materiálu FDM (Fused deposition modeling) 3D tisku. Technologie spočívá ve vytlačování roztaveného termoplastu na sebe skrz žhavicí trysku. Výsledný objekt potom vzniká vrstvením těchto vrstev na sebe.

Pro návrh byl použit program Fusion360 od společnosti Autodesk, který se specializuje na tvorbu 3D modelů, jejich rendering a dokonce i simulace v různorodých aspektech, jako jsou například teplotní nebo mechanické simulace.

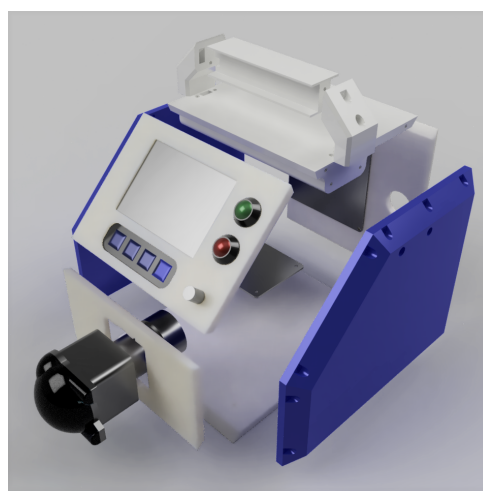
Model byl rozdělen na 8 hlavních částí, aby byl jednoduše vytisknutelný, a aby se minimalizovali náklady při nutnosti úpravy rozměrů.

Model se skládá z:

- Levá a pravá bočnice
- Zadní díl
- Spodní díl
- Díl s výřezem pro pumpu
- Díl pro displej a tlačítka
- Vrchní díl
- Držadlo



(a) Složený model



(b) Rozložený model

Obr. 6.4: Renderý výsledného návrhu zařízení, obrázek (b) je rozložen do prostoru, aby demonstroval spoje mezi jednotlivými tištěnými díly.

Na obrázku 6.4 je ukázka renderu složeného modelu (a) a rozloženého modelu (b). Při návrhu byl brán v potaz i konvenční styl podobných zařízení, zejména nakloněný displej pro lepší čitelnost a přístup k ovládacím prvkům a vrchní držadlo. Držadlo mimo jiné obsahuje i dva výřezy na RGB LED pásy usazené v hliníkové chladicí liště. Tyto pásy budou indikovat stav pumpy a zastoupí světelnou signalizaci pro obsluhu, která bude moci rychle rozpoznat chybové stavy či přijmutí dat z aplikace.

7 Zhodnocení dosažených výsledků

V této kapitole budou zhodnoceny výsledky, kterých tato práce dosáhla. Nejpodstatnější inovace zařízení je právě v bezpečnostní pojistce přetlaku. Běžná zařízení tohoto typu podobné bezpečnostní pojistky neobsahují, a proto pumpa navržená v této práci zvyšuje bezpečnost používání těchto zařízení v provozu.

Další inovací je bezdrátová konektivita, kterou využívá ovládací aplikace a přidává předpoklad pro sdružování vícero takovýchto pump do kooperujících celků.

7.1 Zařízení peristaltické pumpy

Výsledkem práce je reálné zařízení, které implementuje všechny řídicí algoritmy i bezpečnostní ochranu proti přetlaku. Zařízení dokáže přesně dávkovat vodu, což bylo otestováno opakovaným měřením pomocí odměřeného válce. V případě, že se ucpe odvodní trubička, pumpa na to reaguje vypnutím chodu motoru a ohlášením chybového stavu. V zařízení je možné měnit všechny parametry důležité pro chod pumpy. Pumpa se dokáže připojit k MQTT serveru, odesílat i přijímat data a reagovat na ně. Na obrázku 7.1 je porovnání komerční peristaltické pumpy (a) a mnou vyrobeného zařízení (b).



(a) Konvenční pumpa od firmy Shenchon [20]



(b) Fotografie reálného zařízení

Obr. 7.1: Porovnání komerčně prodávaného produktu a výsledného zařízení

7.2 Ovládací aplikace

Počítačová aplikace pro monitorování a vzdálené ovládání pumpy byla navržena velice intuitivně tak, aby korespondovala se vzhledem a ovládáním pumpy. Tím bylo dosaženo snížení náročnosti kladené na uživatele, neboť se eliminovaly problémy s rozdílným ovládáním. Aplikace poskytuje uživateli všechny monitorované údaje z pumpy a dovoluje mu měnit parametry, stejně jako by byl uživatel přímo u pumpy.

7.3 Dosažené výsledky vzhledem k základnímu zadání

Parametry výsledného zařízení se měnily v průběhu jeho vývoje. Porovnání mezi základním zadáním a předpokládanými schopnostmi pumpy je znázorněné v tabulce 7.1, kde je graficky zaznačeno, jaké body byly splněny, jaké byly naopak vynechány a jaké byly vytvořeny nad rámec zadání.

Zadání	Dosažené	Minimální
Podpora typu motoru		
Stejnoseměrný		
Průtok	250 ml/min	100 ml/min
Režimy		
stisk START - zapne pumpu		
puštění START - zastaví pumpu		
START - zapne pumpu		
STOP - zastaví pumpu		
START - odměří přesnou dávku		
START - zapne periodické dávkování		
STOP - zastaví dávkování		
Monitorování		
Proud do motoru		
Otáčky motoru		
Ovládání		
Dotykový displej		
Sedmisegmentový displej		
Rotační enkoder		
Tlačítko START/STOP		
Přepínač ON/OFF		
LED indikující stav pumpy		
LED indikace směru pumpování		
LED indikující periodické dávkování		
Nastavení rychlosti		
Blok funkčních tlačítek		
Konektivita		
Možnost ovládání skrz IoT/MQTT		
USB		
Alarmy		
Ucpání odvodní hadičky		

Tab. 7.1: Tabulka reprezentující parametry minimálního a realizovaného zadání, zelená políčka značí splněné body, bílá políčka značí nesplněné body

Závěr

Začátek práce se zabývá uvedením do tématu a problematiky peristaltických pump, jejich princip funkce, rozdělení a příklady použití. Dále jsou podrobně rozebrány všechny komunikační protokoly a sběrnice, které bude zařízení využívat k svému fungování. Před započítím návrhu samotného zařízení bylo nutné ověřit funkčnost inovativních postupů, kterou mělo výsledné zařízení obsahovat.

Měření proudu do motoru, na němž je postavena premisa bezpečnostního prvku zařízení, potvrdilo očekávané průběhy při přetlaku, ale navíc ukázalo, že je možné měřit i podtlak. Bohužel 24V motor, který byl použit ve výsledné realizaci, byl natolik výkonný, že změna proudu u podtlaku nebyla spolehlivě měřitelná. Měření poskytlo náhled do toho, jak pumpa reaguje na změny toku média, a v jakých mezích se pohybuje proud za daných stavů.

Před výběrem komponent bylo vytvořeno blokové schéma architektury zařízení znázorňující tok informací a signálů mezi jednotlivými bloky pumpy. Po zhodnocení výsledků měření a návrhu architektury byly vybrány komponenty vhodné pro plnění všech zamýšlených funkcí. Z klíčových komponentů byl pro možnosti následného rozšiřování zařízení zvolen procesor ESP32, který má nejen skvělé vlastnosti a vysoký výkon, ale nabízí bezdrátovou konektivitu potřebnou k implementaci bezdrátové komunikace se zařízením. Pro dostatečně přesné měření proudu musel být vybrán senzor s dostatečnou minimální měřitelnou hodnotou. Těmto kritériím vyhovoval senzor INA219. Pro řízení motoru byl zvolen H-můstek s parametry odpovídajícími řízení motoru dle jeho maximálních hodnot s dostatečnou rezervou.

Po shromáždění všech potřebných komponent bylo navrženo schéma zapojení pumpy a to bylo před návrhem desky otestováno na nepájivém poli. Po úspěšných testech byla navržena deska plošných spojů, na kterou byly napájeny všechny komponenty pumpy, čímž vznikl hlavní řídicí modul zařízení.

Díky rešerši trhu byl vytvořen vlastní 3D návrh zařízení. Model zařízení byl úspěšně vytisknut pomocí 3D tiskárny z termoplastu PLA. Do zařízení byla nainstalována veškerá elektronika a mechanika vybraná v kapitole 3.3 a zařízení bylo složeno do jeho celistvé podoby. Za pomoci tmelu, barev a laku byla zvýšena odolnost povrchu a díky velice těsným spojům byla zajištěna i základní odolnost vůči vodě.

Po úspěšném ověření komunikace mezi mikroprocesorem a počítačovou aplikací v kapitole 3.4 byla vyvinuta komplexní aplikace schopná spolehlivě ovládat pumpu a monitorovat data z ní skrz MQTT server. Všechny aspekty aplikace jsou popsány podrobně v kapitole 5.

Výsledkem práce je plně funkční zařízení implementující zamýšlené ochranné prvky a inovativní přístup vzdáleného monitorování a ovládání pumpy.

Literatura

- [1] *Welco: WPX1* [online]. Tokio [cit. 07. 10. 2020]. Dostupné z URL:
<https://www.welco.net/product/wpx1/wpx1_guide.html>.
- [2] *Watson-Marlow pumps* [online]. [cit. 07. 10. 2020]. Dostupné z URL:
<<https://www.watson-marlow.com/sg-en/support/how-do-peristaltic-pumps-work-sanitary/>>
- [3] *Wikipedia: Peristaltic pump* [online]. 17. 12. 2020 [cit. 07. 10. 2020]. Dostupné z URL:
<https://en.wikipedia.org/wiki/Peristaltic_pump>
- [4] *Youtube: Peristaltic pump: recommended accessories* [online]. 19. 4. 2016 [cit. 13. 10. 2020]. Dostupné z URL:
<<https://www.youtube.com/watch?v=Ri8lUTpyr6c>>
- [5] *Allflo: PERISTALTIC HOSE PUMPS* [online]. [cit. 2. 11. 2020]. Dostupné z URL:
<<http://www.allflo.com.au/pump-types/peristaltic-hose-pumps.html>>
- [6] *randolph tubing* [online]. [cit. 2. 11. 2020]. Dostupné z URL:
<<https://www.randolphtubing.com/>>
- [7] *PumpsSystems: Traditional Shoe Roller Designs* [online]. 01. 06. 2014 [cit. 2. 11. 2020]. Dostupné z URL:
<<https://www.pumpsandsystems.com/guidelines-progressive-cavity-peristaltic-pump-maintenance-last-two-parts>>
- [8] *Wikimedia: Peristaltic pump.png* [online]. 26. 02. 2017 [cit. 2. 11. 2020]. Dostupné z URL:
<https://commons.wikimedia.org/wiki/File:Peristaltic_pump.png>
- [9] *AceRoutine: cooperative multitasking library* [online]. Dublin [cit. 21. 5. 2021]. Dostupné z URL:
<<https://github.com/bxparks/AceRoutine>>
- [10] *Vestavné systémy a mikroprocesory* [Skripta]. Autor: Ing. Tomáš Macho, Ph.D. 31. 5. 2020 Brno: VUT [cit. 9. 11. 2020].
- [11] *Medium.com: What is MQTT ?* [online]. Autor: @shashank95 20. 06. 2018 Dublin [cit. 18. 11. 2020]. Dostupné z URL:
<<https://medium.com/@shashank95/what-is-mqtt-e4a6f1a9e483>>
- [12] U. Nanda and S. K. Pattnaik, *Universal Asynchronous Receiver and Transmitter (UART)*, 2016 [cit. 21. 5. 2021].
- [13] *Wikipedia: Universal asynchronous receiver-transmitter* [online]. 8. 5. 2021 [cit. 21. 05. 2021]. Dostupné z URL:
<https://en.wikipedia.org/wiki/Universal_asynchronous_receiver-transmitter>

- [14] *Trade India: S.S. 304 Peristaltic Pump* [online]. [cit. 25. 11. 2020]. Dostupné z URL:
<<https://globalindustriesindia.tradeindia.com/s-s-304-peristaltic-pump-4464907.html>
- [15] *Hackster.io: Azure Sphere and Mikroe Air Quality Sending to IoT Central* [online]. [cit. 4. 12. 2020]. Dostupné z URL:
<<https://www.hackster.io/waltercoan/azure-sphere-and-mikroe-air-quality-sending-to-iot-central-99b156>
- [16] *Sanatbazar.com: Úvod do komunikačných protokolů* [online]. [cit. 4. 12. 2020]. Dostupné z URL:
<https://www.sanatbazar.com/tech/electronic/item/125-%D8%A2%D8%B4%D9%86%D8%A7%DB%8C%DB%8C_%D8%A8%D8%A7_%D9%BE%D8%B1%D9%88%D8%AA%DA%A9%D9%84_%D9%87%D8%A7%DB%8C_%D8%A7%D8%B1%D8%AA%D8%A8%D8%A7%D8%B7%DB%8C_%D8%AF%D8%B1%D8%B3_%D8%B3%D9%88%D9%85_i2c
- [17] *Wikimedia: UART block diagram.svg* [online]. 12. 11. 2018 [cit. 7. 12. 2020]. Dostupné z URL:
<https://commons.wikimedia.org/wiki/File:UART_block_diagram.svg
- [18] Singh, Meena and Rajan, M.A. and Shivraj, V.L. and Balamuralidhar, P., *Secure MQTT for Internet of Things (IoT)* 2015 [cit. 21. 5. 2021].
- [19] Texas Instruments, *INA219 Zero-Drift, Bidirectional Current/Power Monitor With I²C Interface*, SBOS448G 9. 2008 [cit. 21. 5. 2021].
- [20] *Drifton* [online]. [cit. 23. 05. 2021]. Dostupné z URL:
<<https://www.drifton.eu/shop/10-dispensing-tube-pumps/1539-labf6-peristaltic-dispensing-pump/>

Seznam symbolů, veličin a zkratek

SDK	ucelený balíček obsahující vývojářské nástroje k programování – Software Development Kit
OS	operační systém – Operating System
Iot	Internet věcí, trend v programování který, propojuje zařízení přes internet – Internet of Things
RPC	komunikační protokol dotazující se klienta na procedury – Remote Procedure Call
JSON	standardizovaný formát souboru a formát pro výměnu dat používaný jazykem JavaScript který se velmi snadno čte – JavaScript Object Notation
MQTT	komunikační protokol fungující na principu publisher/odběratel – Message Queuing Telemetry Transport
RAM	velice rychlé paměťové čipy – Random-access memory
TLS	zabezpečovací vrstva požadující hesla nebo certifikace k povolení připojení – Transport Layer Security
I2C	komunikační protokol – Inter-Integrated Circuit
SDA	synchronní data – Synchronous Data
SCL	synchronní hodinový signál – Synchronous Clock
ACK	informace o korektním přijetí datového rámce – Accepted
NACK	informace o nekorektním přijetí datového rámce – Not Accepted
UART	komunikační sběrnice – Universal asynchronous receiver-transmitter
mA	jednotka proudu – miliampere
WPM	nejmenší z modelů peristaltických hlav firmy Welco
WPX1	střední model peristaltické hlavy od firmy Welco

Seznam příloh

A Příloha - Komunikace pumpy a aplikace	56
B Příloha - Fotografie zařízení	57
C Obsah přiloženého CD	60

A Příloha - Komunikace pumpy a aplikace

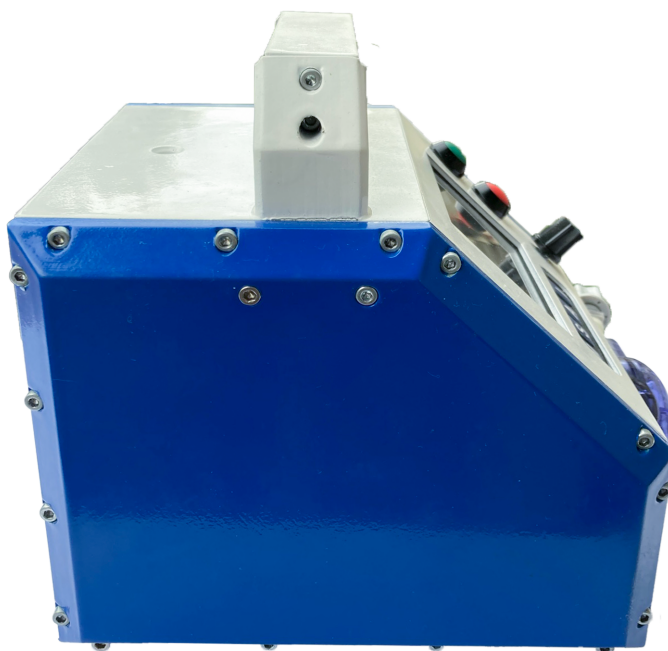


Obr. A.1: Obrázek demonstruje úspěšný přenos dat z pumpy do počítačové aplikace.

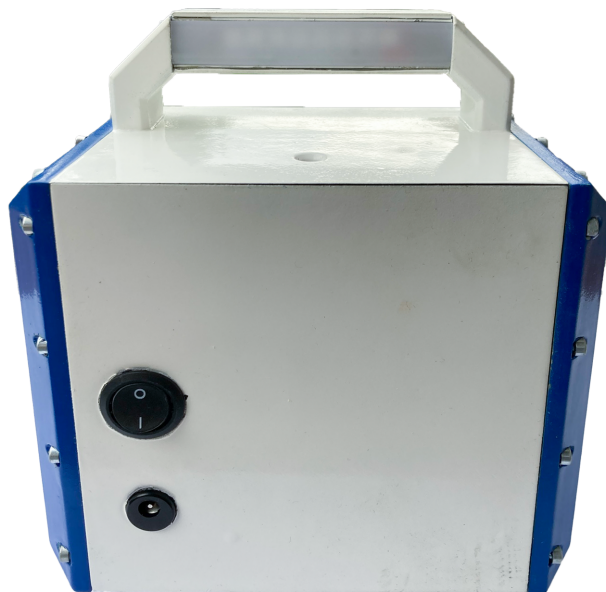
B Příloha - Fotografie zařízení



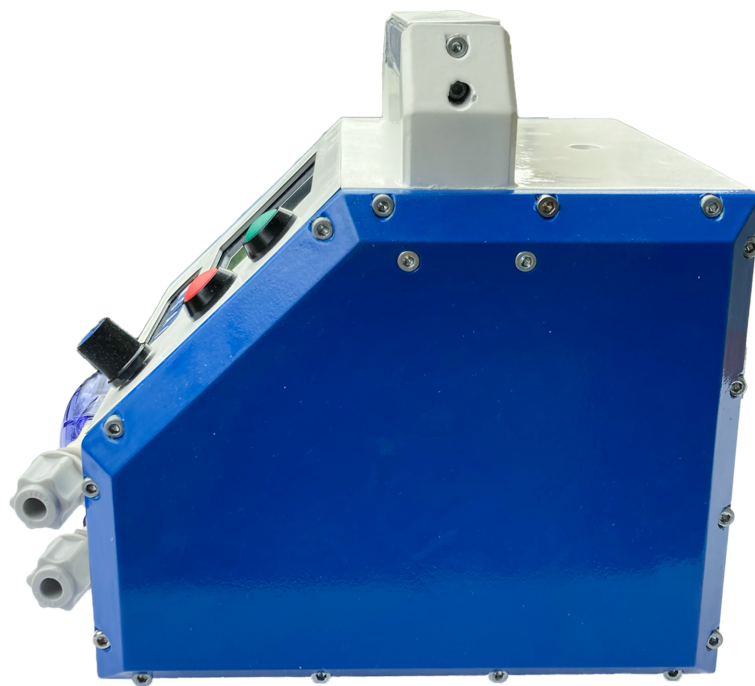
Obr. B.1: Zlepšené Wilsonovo proudové zrcadlo.



Obr. B.2: Zlepšené Wilsonovo proudové zrcadlo.



Obr. B.3: Zlepšené Wilsonovo proudové zrcadlo.



Obr. B.4: Zlepšené Wilsonovo proudové zrcadlo.



Obr. B.5: Zlepšené Wilsonovo proudové zrcadlo.



Obr. B.6: Zlepšené Wilsonovo proudové zrcadlo.

C Obsah přiloženého CD

```
/ ..... kořenový adresář přiloženého CD
├── PumpFirmware ..... složka se soubory firmwaru zařízení
│   ├── .vscode
│   ├── include
│   ├── lib ..... složka se soubory použitých knihoven
│   ├── src ..... zdrojová složka projektu
│   │   ├── Alarm ..... složka se soubory třídy Alarm
│   │   ├── Controls ..... složka se soubory třídy Controls
│   │   ├── Display ..... složka se soubory třídy Display
│   │   ├── Mqtt ..... složka se soubory třídy Mqtt
│   │   ├── Pump ..... složka se soubory třídy Pump
│   │   ├── main.cpp ..... zdrojový soubor projektu
│   │   ├── mainLib.cpp ..... zdrojový soubor knihovny mainLib.h
│   │   └── mainLib.h ..... hlavní knihovna projektu
│   └── platformio.ini ..... nastavení procesoru ESP-32
├── PumpAppSoftware ..... složka se soubory softwaru aplikace
│   ├── bin ..... zdrojová složka projektu
│   │   ├── Debug ..... kompilační složka projektu
│   │   └── PeristalticApp.exe ..... spustitelný soubor aplikace
│   ├── Classes ..... složka obsahující třídy z projektu
│   ├── Pages ..... složka obsahující zdrojové soubory stránek projektu
│   ├── MainWindow.xaml ..... grafické rozhraní první vrstvy aplikace
│   ├── MainWindow.xaml.cs ..... zdrojový kód první vrstvy aplikace
│   └── PeristalticApp.sln ..... projekt Visual Studio
└── Přípravek pro řízení peristaltické pumpy, Jakub Charvát, 2021.pdf
```